GIUGNO 1984 L. 10.000

TI99 NEWSOFT

SOFTWARE & Software

TI99 - RIPRENDIAMOCI IL COMPUTER

8 SUPERPROGRAMMI SU CASSETTA

BASIC - EXTENDED: I TRUCCHI, I MISTERI

ASSEMBLER E L/M

SOMMARIO

☐ **«DISPLAY AT» NEL TI 99**
Come si possono visualizzare i caratteri in qualsiasi posizione dello schermo. In TI BASIC. pag. 3

☐ **PRIMI PASSI CON L'ASSEMBLER DEL TI 99/4A**
Un mondo straordinario, un modo nuovo per programmare il TI 99. In questo numero: GRAPHICS 1. pag. 4-5

☐ **QUATTRO CHIACCHIERE TRA NOI**
Un programma di statistica per tutti, insieme ad un artico di facile comprensione. pag. 5-6

☐ **OTHELLO**
Un gioco di strategia per due persone in versione superspeciale per il nostro TI 99 pag. 6-7

☐ **INCONTRI RAVVICINATI**
Colori, suoni e memoria. Una versione del gioco "Simon" da giocare contro il TI 99. pag. 8

☐ **CARATTERI IN CORSIVO**
Un intero set di caratteri in corsivo. Sulla cassetta il programma in EXTENDED, sulla rivista quello in BASIC. pag. 9

☐ **GALAXY II**
Una emozionante battaglia spaziale in TI BASIC. pag. 10-11

☐ **CUBO MAGICO**
Armatevi di pazienza e cercate di risolvere questo cubo di Rubik. pag. 12

☐ **NIBBLER**
Imperversa nelle sale gioco e nei bar. È il serpentone che non deve mangiarsi la coda. pag. 13

☐ **DIFESA**
Non è il "Parsec", ma gli arriva vicino. Difendetevi dalle astronavi attaccanti. pag. 14-16

TI 99... RIPRENDIAMOCI IL COMPUTER!

Aprite gli armadi, rifate il percorso del « dimenticatoio », scendete in cantina. Cercate di ricordarvi dove l'avete nascosto, perché è finalmente arrivato il momento di tirare fuori il vostro, nostro, *VECCHIO* **TI 99**.
Certamente molti di voi il TI 99, non l'hanno mai messo da parte, ma quelli che invece dopo l'euforia del « primo momento » lo hanno fatto, potranno contare su di **noi** (e specialmente su di VOI), per *innamorarsi* sempre di più del TI 99.
Magari adesso vi state chiedendo NOI chi siamo.
Ma **TI 99-NEWSOFT**, no! La prima, l'unica rivista italiana solo e soltanto per il TI 99.
Se ci state leggendo, significa che avete acquistato la rivista, e, sorpresa, TI 99 NEWSOFT contiene una cassetta.
È una scelta, questa, che abbiamo fatto dopo aver chiesto a molti amici 99viani, se una cassetta, contenente tutto il software illustrato nella rivista, avrebbe incontrato il loro favore.
All'unanimità, la risposta è stata SI! Niente più errori nei listati, ore faticosissime di trascrizione sul 99 e arrabbiature perché il programma non gira.

Insomma *TI 99 NEWSOFT* **è una rivista...** con cassetta,
due cose diverse e complementari.
Nella rivista (che avrà sempre più pagine) vi daremo sempre articoli,
consigli, informazioni sul nostro TI 99. Non dimenticheremo
l'hardware, il « fai da te » e cento altre proposte interessanti per il
profano e per chi crede di sapere già tutto del TI 99.
Nella *cassetta* troverete sempre tanto software originale:
per giocare, lavorare e studiare.
Ma quel che più conta è che TI 99 NEWSOFT vogliamo sia la
VOSTRA rivista per il VOSTRO computer. Quindi niente paura!
Se avete idee, se avete voglia di scrivere, se volete pubblicare
un vostro pragramma, se volete vendere i vostri programmi,
se volete scambiare idee e opinioni con gli altri « 99viani »
NOI siamo qui.
Fatevi avanti senza alcun timore, perché abbiamo spazio
per tutti e tutti noi vogliamo « crescere insieme ».
E ricordatevi sempre che il nostro TI 99 non è un computer
da quattro soldi (anche se forse l'avete pagato una « miseria »)
e non è stato ancora SEPOLTO.
Siamo qui per dimostrarvelo.

A.L.

«DISPLAY AT» NEL TI 99

Molti affermano che imparare a programmare in BASIC su piccole macchine sia tutt'altro che positivo, questo non sempre è vero. A tutti gli effetti la tecnica di programmazione rimane invariata, anche se notevolmente facilitata da apposite istruzioni.

La TEXAS INSTRUMENTS, per rendere più competitivo il suo computer, ha pensato di implementare il TI 99/4a con un tipo di BASIC standard, molto simile, fatta eccezione per le istruzioni grafiche e sonore, al MICROSOFT.
Questo tipo di linguaggio è molto usato anche su macchine di più grandi dimensioni, addirittura i programmi scritti su questi computers sono completamente, o quasi, compatibili con il nostro home.
Qui iniziamo una rubrica, esclusivamente riservata al TI BASIC ed al TI EXTENDED, dove tenteremo di spiegare nel miglior modo tutti quegli argomenti che sul manuale non sono sufficentemente chiari.
Dopo qualche discussione, in redazione abbiamo deciso di riunire le due rubriche, TI BASIC e TI EXTENDED, , in quanto, trattandosi di due linguaggi strettamente legati tra loro, avremmo potuto correre il rischio di ripetere più volte gli stessi concetti.
Cercheremo inoltre, nei limiti del possibile, di realizzare dei piccoli programmi per rendere più assimilabili i due linguaggi.
Iniziamo questa volta con la routine «DISPLAY AT».
Essa consente di visualizzare dei caratteri in qualsiasi posizione dello schermo, offrendo, anche a tutti gli utenti che non posseggono il modulo applicativo TI EXTENDED BASIC, la possibilità di usufruire di questa utility.

DESCRIZIONE DEL PROGRAMMA:

Le istruzioni usate sono solo due, una FOR-NEXT, ed una CALL HCHAR.
Per poter utilizzare questo semplice programma, dovete capire a fondo l'utilizzo delle tre variabili usate: due sono numeriche ed una di stringa.
Nelle variabili numeriche dobbiamo inserire rispettivamente la riga e la colonna dove desideriamo sia visualizzato il messaggio...
Ovviamente, la variabile di stringa deve contenere la frase.
Per comodità abbiamo strutturato il programma in maniera tale da permettervi di inserire tutte le informazioni (nell'ordine: riga, colonna, stringa) in una stessa stringa.
Ricordatevi che se volete indicare i numeri da 1 a 9, dovete aggiungere lo 0 (zero) prima del numero.
Es.: riga = 9 colonna = 5
stringa = CIAO
Variabile di
stringa = "0905CIAO"
Per una maggiore comprensione del programma pubblichiamo una versione espansa, ma per ottenere una maggiore velocità di esecuzione alleghiamo anche la versione compressa.

(S.T. M.E.)

ROUTINE DISPLAY AT espansa

```
10 CALL CLEAR
20 A$ = "1206NEW SOFT
FOR TI-994A"
30 GOSUB 10000
•
•
•
10000 REM DISPLAY AT
10010 RIGA = VAL
(SEG$(A$,1,2))
10020 COLO = VAL
(SEG$(A$,3,2))
10030 FOR LOOP = 5 TO
LEN(A$)
10040
A = ASC(SEG(A$,L,1))
10050 CALL HCHAR
(RIGA,COLO,A)
10060 C = C + 1
10070 NEXT LOOP
10080 RETURN
```

ROUTINE DISPLAY AT compressa

```
5 C = 0
10 CALL CLEAR
20 A$ = "1206NEW SOFT
FOR TI-994A"
30 GOSUB 10000
•
•
•
10000 REM DISPLAY AT
10010 FOR L = 5 TO
LEN(A$)
10020 CALL HCHAR(VAL
(SEG$(A$,1,2)),VAL(SEG$
(A$,3,2,)) + C,ASC(SEG$(A$,
L,1,)),1)
10025 C = C + 1
10030 NEXT L
10040 RETURN
```

PRIMI PASSI CON L'ASSEMBLER DEL TI-99/4A

Con una serie di articoli dedicati a queso linguaggio, non vogliamo certamente proporvi un corso di apprendimento rapido dell'assembler, ma cercheremo di aiutare tutti gli utenti del «TI» che, come noi, volendo sfruttare le straordinarie possibilità del TMS 9900, si sono trovati di fronte a parecchie difficoltà, prima fra tutte la scarsa documentazione.
Ci sarà da parte nostra il tentativo di colmare, almeno in parte, queste lacune, e cercheremo di farlo in modo scorrevole, cioè alternando descrizioni tecniche con semplici esempi di programmazione.
Prima di poter parlare di ASSEMBLER, bisogna specificare che in questo caso il sistema in configurazione base non è sufficiente, è necessario ricorrere ad una delle due seguneti configurazioni hardware:
- 1 - MODULO SSS
Mini Memory
- 2 - MODULO SSS
Editor/Assembler, Unità a dischi, Espansione 32K Ram
È evidente che il sistema più pratico è il primo, in quanto permette di raggiungere lo scopo con una spesa decisamente inferiore a quella necessaria per raggiungere la seconda configurazione.
L'Editor/Assembler permette, però, di ottenere maggiori facilitazioni nella scrittura e nell'elaborazione di programmi assembler.
Fate queste premesse, armiamoci di buona volontà e addentriamoci in questo affascinante mondo.

PER INCOMINCIARE

Il primo argomento che affrontiamo si riferisce alla descrizione dettagliata del «Video Display Processor» (TMS 9929 per la versione europea).
Uno dei pregi del «VDP», oltre all'alta risoluzione grafica (256*192 pixel) è rappresentato dagli sprites.
Gli sprites sono delle figure con una risoluzione grafica di 8*8 pixel oppure di 16*16 pixel, a seconda delle necessità.
Essi sono indirizzati sul video mediante coordinate (0-256 in orizzontale e da 0-192 in verticale) e si sovrappongono all'immagine esistente, senza alterarne il contenuto.
Contrariamente al TI EXTENDED BASIC, che può gestire solo 28 sprites, l'assembler può visualizzarne contemporaneamente 32, con la sola limitazione di non metterne più di quattro su una stessa linea orizzontale.
Un'altra caratteristica interessante del «VDP» è data dalla possibilità di poter scegliere liberamente tra quattro modi di rappresentare dell'insieme dei patterns:
- 1 - GRAPHICS 1
- 2 - GRAPHICS 2
- 3 - TEXT
- 4 - MULTICOLOR

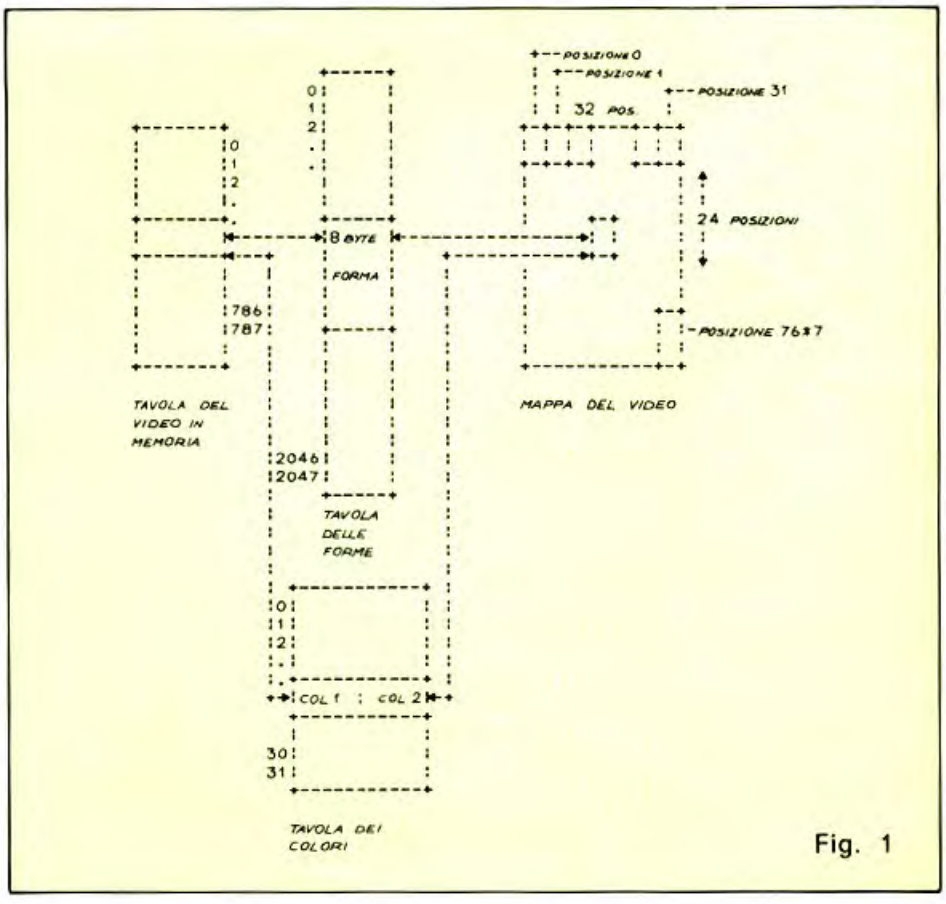


Fig. 1

· *GRAPHICS - 1*

In questo modo grafico lo schermo è diviso in una griglia di 32 colonne per 24 righe dove possiamo visualizzare un carattere.
Ogni carattere è costituito da 8*8 pixel cioè 64 bit, ed è rappresentato nella griglia con un numero, «nome del carattere», compreso tra 0 e 255.
Ad ogni «nome» corrisponde una matrice di 8 byte nella tavola delle forme.
Oltre alla tavola video e a quella delle forme, esiste una tavola dei colori che è composta da 32 byte, ognuno dei quali definisce il colore interno ed esterno di uno degli insiemi di caratteri «nomi».
Per insieme si definisce un gruppo di 8 nomi.
Per ulteriori chiarimenti vedere la «figura 1».

(continua, 1)
(S.T.-M.E.)

PICCOLO GLOSSARIO PER CAPIRCI MEGLIO

PATTERN: forma che definisce un carattere.
PIXEL: il più piccolo punto definibile sullo schermo
BIT: la più piccola informazione riconoscibile da un circuito digitale.
BYTE: insieme di 8 bit.
HARDWARE: termine che identifica tutte parti meccaniche ed elettroniche di un calcolatore.
SOFTWARE: insieme di programmi che permettono, insieme con l'ARDWARE, di rendere utilizzabile un computer.

QUATTRO CHIACCHIERE TRA NOI

Avrete senz'altro sentito dire spesso che gli Home Computers hanno due destini.
Il primo è quello di fare concorrenza ai videogiochi, riducendosi al ruolo di esecutori di avventure tutte uguali, basate, per la maggior parte, cul cliché «ammazza il mostro e prendi i soldi».
Il secondo destino è quello di andare a fare compagnia al baule dei vestiti vecchi in soffitta.
È una opinione, forse da voi non condivisa, ma dovrete riconoscere che, in fondo, può esserci un pochino di verità.
Confessiamolo! Tutti noi ci siamo domandati, passata l'iniziale euforia provata nel momento dell'acquisto, quale può essere una forma adeguata di utilizzo del nostro TI 99.
Ebbene, uno degli impieghi più naturali per un calcolatore è quello di «eseguire i calcoli»; non quelli, naturalmente, che è più agevole eseguire su quelle macchinette da 20.000 Lire che hanno ormai invaso le nostre case, ma piuttosto quel tipo di operazioni lunghe, ripetitive che, se fossimo costretti ad eseguire di persona, molto probabilmente ci condurrebbero alla nevrosi.
Per coloro che hanno necessità di controllare dati, di verificare misure, o soltanto per coloro a cui potrebbe interessare sentire, così alla buona, rudimentali nozioni di quella disciplina scientifica che è la *statistica*, pubblichiamo un programma di calcolo che, riteniamo ai più possa essere utile e che serve da introduzione a calcoli più complessi che certamente il nostro TI 99 è in grado di fare.

DUE PAROLE SEMPLICI, DI STATISTICA

Ad alcuni forse la parola «statistica» può generare sconforto. Ad altri il pensiero potrebbe subito andare alla famosa storia del pollo: la sapete? Bene. Ma per i due o tre che non la conoscono, la riassumeremo brevemente.
Il paeta romano Trilussa, in un sonetto, dice pressapoco che: la statistica è quella «cosa» che fà sì che, se tra due persone, una mangia due polli e l'altra nemmeno uno, afferma che è stato mangiato un pollo a testa.
Bravo Trilussa, ma il concetto è sbagliato.
Il concetto del pollo a testa è una tendenza di media ed è proprio la statistica che ci permette di verificare questo dato medio, di verificarne l'attendibilità e misurarne la dispersione.
Il compito della statistica è quello di ottenere informazioni su una certa quantità di dati, descrivere la tendenza centrale al raggruppamento di questi dati e la dispersione intorno ad essa.

LA MEDIA ARITMETICA

Supponiamo di avere una serie **n** di dati
X1,X2,X3,...Xn, la media di essi sarà data da

X1 + X2 + X3 + ...Xn/n; nulla da eccepire? Sì qualcosa: la media sarà certamente una misura esatta, ma cadiamo nella questione del pollo; non sappiamo ancora se qualcuno è rimasto a bocca asciutta, o se qualcuno ne ha mangiati due o tre.

LA MEDIANA

Mettiamo ora i valori **n** di prima in un certo ordine discendente (prima il più grande) o ascendente (per primo il più piccolo, poi l'altro, e così via), e poi scegliamo il valore di mezzo (se per caso la lista è costituita da un numero pari di dati, nessun problema, prendiamo i due che stanno nel mezzo e facciamo la media di essi). Per inciso, se vi sono difficoltà nel mettere in ordine questi dati perché sono troppo numerosi, esiste pur sempre il nostro valoroso TI 99. Questo tipo di ordinamento è chiamato Sort e il nostro computer ci verrà in aiuto con programmi di Sort che decriveremo più avanti.
Cosa otteniamo con la misura della mediana? Un valore che è meno influenzato da grosse sproporzioni tra i dati.
Facciamo un esempio: 9 persone hanno 1000 lire, Caio, il decimo, ne ha 1.000.000. Possiamo pensare di andare tutti in ristorante di lusso e abbandonarci in canti e libagioni? Si, se paga Caio, altrimenti noi nove laviamo i piatti per una vita.
La misura della media ci dice che possediamo 109.000 lire a testa; la realtà, la mediana, modestamente ci avverte che abbiamo 1000 lire. Forse, bastano per un panino. Caio è un godereccio e se la cava da solo.

LA DEVIAZIONE STANDARD

D'accordo Caio se la cava da solo, abbastanza bene del resto, ma qualcuno dovrà pur accorgersi di questa ingiustizia, e di quella del signore dei due polli del sonetto di Trilussa.
Giustizia vuole che esista la *deviazione standard,* una funzione che ci avverte, per esempio, della disomogeneità della distribuzione, della sperequazione esistente tra noi e Caio, se volete, in Italia, tra noi e... lasciamo perdere.
Se esprimiamo con **u** la media dei valori, possiamo dire che la deviazione standard **s** sarà uguale alla radice quadrata di:
$((\times 1\text{-}u)^2+(\times 2\text{-}u)^2+(\times 3\text{-}u)^2$ ecc.)/n (n = numero totale dei dati).
Provate a calcolare qualche volta, con il programma che troverete nella cassetta alcune deviazioni, vedrete che più alto sarà il valore ottenuto, più alta risulterà la dispersione dei valori che avete dato.
E per ora fermiamoci qui. In seguiro potremo tentare di applicare quanto abbiamo appreso.

(P.C.)

OTHELLO

Othello è un gioco di strategia di ottimo livello che si gioca tra due persone. Ricorda l'antico REVERSI e, nonostante sia alla portata di tutti per le sue semplici regole, è molto stimolante per le difficoltà e le possibilità strategiche che offre.

Il gioco si svolge su di una scacchiera di 8 caselle per lato; ogni giocatore ha a disposizione una serie di pedine con una faccia di un colore ed una di un altro. Alla partenza sulla scacchiera sono già disposte quattro pedine, due girate su un colore e due sull'altro.

A questo punto inizia il gioco che consiste nel porre a turno una pedina del proprio colore sulla scacchiera, in modo che sia adiacente ad una di colore diverso e che, sulla stessa riga o colonna o diagonale sia presente un'altra pedina del proprio colore. Si tratta in pratica di conquistare una o più pedine avversarie comprendendole tra due proprie. È obbligatorio conquistare almeno una pedina avversaria, altrimenti si cede la mossa.
La partita termina quando saranno state deposte tutte le 64 pedine o quando nessuno dei due giocatori può muovere; il vincitore sarà colui che avrà più pedine del proprio colore sulla scacchiera.

IL GIOCO

Il programma può funzionare sia in TI BASIC, sia in EXTENDED BASIC, e può accettare i joystick, e i comandi da tastiera.
Il computer richiederà per prima cosa i nomi dei due giocatori; se uno di questi è COMPUTER il gioco si svolgerà allora tra voi e la macchina. È anche possibile fornire due volte il nome COMPUTER ed in questo caso la macchina giocherà in modo autonomo contro sé stessa.
Secondo le regole precedenti esposte il giocatore dovrà segnalare la casella scelta per porre la propria pedina muovendo i 2 cursori visibili sul lato superiore e sul lato sinistro della scacchiera con la leva di uno dei due telecomandi, oppure con i tasti contrassegnati dalla FRECCE, «E» «X» «S» «D»; quando il puntino nero si sarà posizionato sulla casella scelta si potrà effettuare il tiro premendo il pulsante di sparo o la barra spaziatrice della tastiera.
Se il tiro non è valido, perché non conquista nessun segnalino, verrà emessa una segnalazione di errore e lo si potrà ripetere. Se non esiste nessuna casella che permetta la conquista di pedine avversarie si dovrà rinunciare al tiro premendo il tasto «R». In qualsiasi momento del gioco si potrà ripartire dall'inizio premendo il tasto «I», oppure chiudere la partita premendo il tasto «B».
Durante la partita sullo schermo saranno indicati il nome del giocatore che deve muovere, il segnalino del suo cursore e i contatori delle pedine deposte. Al termine verrà indicato il vincitore.

CARATTERISTICHE DEL PROGRAMMA

LISTATO

Linee	descrizione della funzione
10- 380	Inizializzazione delle variabili e presentazione titolo
390- 510	Caricamento della tabella mossa
520- 660	Definizione caratteri speciali
670- 760	Richiesta dei nomi dei giocatori
770- 860	Costruzione del quadrante di gioco
870-1130	Sistemazione dei primi quattro segnalini
1140-1200	Completamento della videata di partenza
1210-1380	Richiesta della mossa da effettuare. Se il tiro spetta al computer viene richiamata automaticamente l'opportuna routine
1390-1620	Spostamento del cursore orizzontale a destra o sinistra
1630-1860	Spostamento del cursore verticale in alto o in basso
1870-2230	Primo controllo di validità della mossa con eventuale segnalazione di errore
2240-2290	Rinuncia della mossa
2300-2650	Secondo controllo di validità della mossa e rovesciamento dei segnalini conquistati
2660-3270	Strategia del computer
3280-3310	Routine di display di scritte orizzontali
3320-3350	Routine di display dei contatori
3400-3740	Routine di incremento contatori e controllo di fine partita
3750-3780	Routine di display di scritte verticali
3790-3840	Routine di display del giocatore che deve effettuare la mossa
3850-3910	Routine per il controllo di alcune caselle chiave durante il tiro del computer

VARIABILI

GIO$(2) ..... contiene i nomi dei due giocatori (dei quali 1 può essere COMPUTER)
MOSSE(7,16) ..... matrice delle mosse in ordine di importanza
QUA(8,8) ..... matrice con la situazione dei segnalini
TIRI ..... contatore del numero dei tiri effettuati
SW ..... switch impostato dopo aver trovato almeno un segnalino avversario da conquistare
SW1 ..... switch indicante la possibilità di conquistare segnalini
SWR ..... switch impostato in caso di rinuncia alla mossa
SWCH ..... switch per il rovesciamento dei segnalini
SWOK ..... switch indicante la validità della mossa
I,L,R,C,RIG,J,K .... variabili temporanee di comodo
G ..... campo indicante il turno di gioco
COM$ ..... contiene la stringa per il display sul video
NOM$ ..... contiene temporaneamente un nome di giocatore
K,S ..... rispettivamente chiave e status della tastiera nelle CALL KEY
C1,C2 ..... contatori dei segnalini
CHA ..... codice ASCII del carattere letto sullo schermo
RIGA,COL ..... coordinate per la scansione del quadrante
CTR,CC ..... contatori parziali dei segnalini conquistati nella mossa in corso
RC,CC ..... variabili temporanee contenenti le coordinate di una mossa del computer
IC,LC ..... coordinate temporanee per la matrice delle mosse del computer
RR1,CC1 ..... coordinate delle caselle chiave

(A.S.)

INCONTRI RAVVICINATI

Non fatevi ingannare dal titolo, non è una versione computerizzata dell'omonimo film!

Chi di voi non ha mai giocato all'ormai famosissimo gioco di «SIMON»!

Questa è senza alcun dubbio la migliore versione per il TI-99/4A.

Per la sua grafica e per i suoi effetti sonori, merita particolare attenzione.

IL GIOCO

Si tratta come forse avete già capito di un gioco musicale, un'alternativa alle ormai troppo impegnative e a volte banali battaglie spaziali.

Il computer di volta in volta proporrà una sequenza di note sempre più lunga che voi dovrete memorizzare e ripetere (ovviamente utilizzando la tastiera).

Per facilitarvi, ogni nota sarà associata ad un colore.

Attenzione il tempo è limitato, e la sequenza metterà sempre più a dura prova la vostra memoria.

Non fatevi spaventare dalla quantità di tasti utilizzati dal gioco, perché il loro utilizzo è relativamente semplice.

Analizzeremo ora singolarmente la funzione dei vari tasti:

tasto	spiegazione
1 -	ripete le note sino a quel momento introdotte
2 -	cancella l'ultimo dato inserito
3 -	vi fa ascoltare una sequenza di 50 note musicali
4 -	aumenta la velocità di esecuzione della sequenza
5 -	non vengono visualizzati i colori
6 -	per finire
B -	corrisponde al colore blu
Y -	corrisponde al colore giallo
R -	corrisponde al colore rosso
W -	corrisponde al colore bianco
G -	corrisponde al colore verde

Non arrendetevi facilmente, perché *ad ogni 50 punti, una sorpresa sempre diversa vi attende...* BUONA FORTUNA...

(E.M.)

CARATTERI IN CORSIVO

È un semplice programmino che vi permetterà di ottenere un intero set di caratteri in corsivo.

Certo avrete notato che una delle cose che mancano sul nostro TI 99/4a è proprio questa. Il nuovo set di caratteri viene sovrapposto a quello delle minuscole standard comprendendo i codici da ASCII 97 a ASCII 122, apici compresi.

Senza troppa difficoltà si può convertire il programma per il TI BASIC, cosicché anche i meno fortunati, possano usufruirne.

Comunque abbiamo preparato la conversione in TI BASIC del programma inciso sul nastro (vedi riquadro).
Vi ricordiamo che il nuovo set di caratteri rimane visualizzato soltanto in fase di esecuzione di un programma «ovviamente BASIC».
Per i fornunati possessori del TI EXTENDED, esiste un metodo per far rimanere invariati i patterns «caratteri», basta far credere all'interprete BASIC di essere ancora in modo esecutivo, ma, come dice un vecchio proverbio: tra il dire e il fare c'è di mezzo il mare. Ma niente paura, con poche righe di programma potremo ottenere il risultato voluto.
Scrivere in modo comando le seguenti istruzioni:
CALL CHAR(97,"«stringa della lettera A in corsivo»")::ACCEPT Y
premete ENTER e FCTN clear.
Attenzione se commettete un «SINTAX ERROR» o un qualunque altro errore, l'interprete cambierà il vostro carattere con quello originale.
Consigliamo di utilizzare questa routine come implementazione di programmi gestionali o di word processing, perché le minuscole così ridefinite permettono una migliore lettura e comprensione dei testi.
Nessuno però vi vieta di usarla per altri scopi!

(S.T.)

CARATTERI IN CORSIVO IN TI BASIC

```
100 DATA 0000003848483400,0040407048483000,0000003840403800,0008083848483400,
0018243C201800,0018282038202000
110 DATA 0000384848380830,0020203824242400,0010003010103800,0010003010105020,
00405060504800,0030101010103800
120 DATA 0000006C54545400,0000007848484800,0000003048483000,0000705848704040,
0038484838080 8,0000002830202000
130 DATA 0000182038083000,0010103810101800,0000004848483400,0000002424281000,
00004454542800,0000002810282800
140 DATA 0000004848301020,0000003810203800
150 RESTORE 100
160 FOR ZO=97 TO 122
170 READ Z$
180 CALL CHAR (ZO,Z$)
190 NEXT ZO
```

GALAXY II

Si tratta di uno dei quei giochi convenzionalmente chiamati spaziali. Nessun'altra definizione può essere infatti più calzante per classificare questo accattivante e dinamico passatempo. Certo, il numero che completa il nome presuppone l'esistenza di un Galaxy 1, e difatti suo fratello maggiore (in quanto progettato prima) ha fornito gli spunti e le routine-base, che hanno permesso la «nascita» del suo ben più articolato e rifinito successore.

IL GIOCO

La situazione di gioco vi pone al comando di una potente astronave dotata di veloci missili teleguidati (da voi naturalmente), che strenuamente cerca di bloccare l'ennesima invasione aliena. Gli extraterrestri in questione, che pilotano due navicelle, hanno il compito di atterrare ed impadronirsi del pianeta da cui eravate decollati, o almeno riuscire ad eliminarvi, lanciandovi contro missili su missili. A dar loro manforte, dal pianeta, un cannone laser tenterà di cogliervi sulla sua traiettoria per disintegrarvi. Per ogni astronave nemica abbattuta verrà visualizzato un teschietto che segnalerà il vostro successo. Una volta che ne avrete collezionati venti, potrete col braccio indolenzito, congratularvi con voi stessi per la buona riuscita della missione. Come ciliegina sulla torta (e per non essere facili prede) i vostri «avversari» proietteranno solo una loro immagine, pura apparenza che ha lo scopo di confondervi; la vera posizione quindi sarà più in basso di quello che l'ombra potrebbe far credere. Come se non bastasse, ai nemici subito individuabili, un altro se ne aggiunge, più subdolo e silenzioso: il tempo.
Visualizzato in basso sullo schermo, di colore verde, trascorre con velocità variabile a seconda dell'opzione scelta nella prima pagina vi-

LISTATO

Linee	descrizione della funzione
220-320	Schermata principale con lista opzioni
330-370	CALL KEY scelta difficoltà
380-420	Variabili controllo difficoltà
430-540	Assegnazione variabili principali
550-730	Assegnazione colori e grafica
740-1070	Costruzione paesaggio
1080-1120	Controllo tempo
1130-1250	Posizioni iniziali di gioco
1260-1400	Stampa scritte
1410-1490	Sparo e azioni cannone alieno
1500-1620	Movimenti e sparo astronave aliena 1
1630-1750	Movimenti e sparo astronave aliena 2
1760-1890	CALL KEY e CALL JOYST per spostamenti e sparo astronave del giocatore
1900-2040	Spostamento a destra dell'astronave del giocatore
2050-2120	Spostamento a sinistra dell'astronave del giocatore
2130-2270	Spostamento in basso dell'astronave del giocatore
2280-2350	Spostamento in alto dell'astronave del giocatore
2360-2450	Sparo dell'astronave del giocatore
2460-2480	Stampa scritta relativa all'atterraggio delle astronavi aliene
2490-2620	Suoni e controlli in caso di abbattimento di un'astronave aliena
2630-2640	Stampa scritta relativa alla vittoria del giocatore
2650-2670	Richiesta di un'altra partita
2680-2740	Suoni e controlli in caso di abbattimento dell'astronave del giocatore
2750-2780	Controllo del numero di astronavi ancora integre
2790-2800	Stampa relativa all'esaurirsi del tempo a disposizione

deo che si presenta dopo aver dato il RUN. Una volta scelto il livello di difficoltà più consone alla vostra abilità, si entra nel vivo del gioco, il cui svolgimento dovrebbe a questo punto essere chiaro. Per muovervi potete usare sia la tastiera sia il joystick. Nel primo caso i tasti da usare sono E (su), X (giù), S (sinistra), D (destra) e «.» (punto) per sparare; nel secondo caso, invece, ricordatevi di sbloccare il tasto ALPHA LOCK. Crediamo di aver detto tutto; ora non vi resta altro da fare che caricare il programma e giocare.

(M.S.)

VARIABILI

A = Riga per la visualizzazione relativa all'astronave aliena 1
B = Colonna relativa all'astronave aliena 1
C = Riga per la visualizzazione relativa all'astronave aliena 2
D = Colonna relativa all'astronave aliena 2
E = Numero di teschietti che vanno visualizzati
F = Riga per la visualizzazione relativa al cannone alieno
G = Numero di astronavi che rimangono al giocatore
I = Variabile di casualità di sparo dell'astronave 2
L = Riga per la visualizzazione relativa alla prima serie del paesaggio
M = Riga relativa alla seconda serie di caratteri per il paesaggio
N = Riga relativa alla terza serie di caratteri per il paesaggio
O = Variabile di casualità di sparo del cannone alieno
P = Variabile di casualità di sparo dell'astronave 1
Q = Riga per la visualizzazione relativa al numero di teschietti
R = Variabile contenente il livello di difficoltà
S = Variabile di Status nelle CALL KEY'
T = Variabile contenente il tempo rimanente
U = Variabile contenente il numero di ufo alieni distrutti
V = Colonna per la visualizzazione relativa del proiettile dell'astronave 2
W = Colonna relativa del proiettile dell'astronave 1
Z = Carattere di controllo del proiettile letto nelle CALL GCHAR
K = Colonna relativa del proiettile dell'astronave del giocatore
X = Riga per la visualizzazione relativa all'astronave del giocatore
Y = Colonna relativa all'astronave del giocatore
AA = Variabile di ritorno nel controllo dei joysticks (movimento 1)
BB = Variabile di ritorno nel controllo dei joysticks (movimento 2)
HH = Variabile di ritorno nel controllo dei joysticks (sparo)
KEY = Variabile di ritorno nel controllo da tastiera

MAGICO CUBO

Chi non ricorda il diabolico Cubo di Rubik, quella struttura apparentemente innocua che qualche anno fa imperversava in tutta l'Europa? Si trattava, come ben ricorderete, di ricostruire le facce del cubo secondo colori uniformi, dopo averlo a malincuore mescolato (qualcuno doveva regolarmente ricorrere all'amico esperto per poter riavere il «suo» cubo in ordine!). Complicati criteri di matematica vettoriale cercarono di spiegare allora la risoluzione del problema del diabolico inventore, ma non fecero altro che complicare inutilmente le cose, trasformando un gioco in un problema da ingegneri. In realtà le sei facce del cubo possono essere rappresentate perfettamente sullo schermo del televisore con il vostro TI 99/4A: in questo modo la ricerca delle mosse avviene in maniera molto più naturale e veloce, e non impazzirete nel tentativo di ricomporre una sola misera faccia!

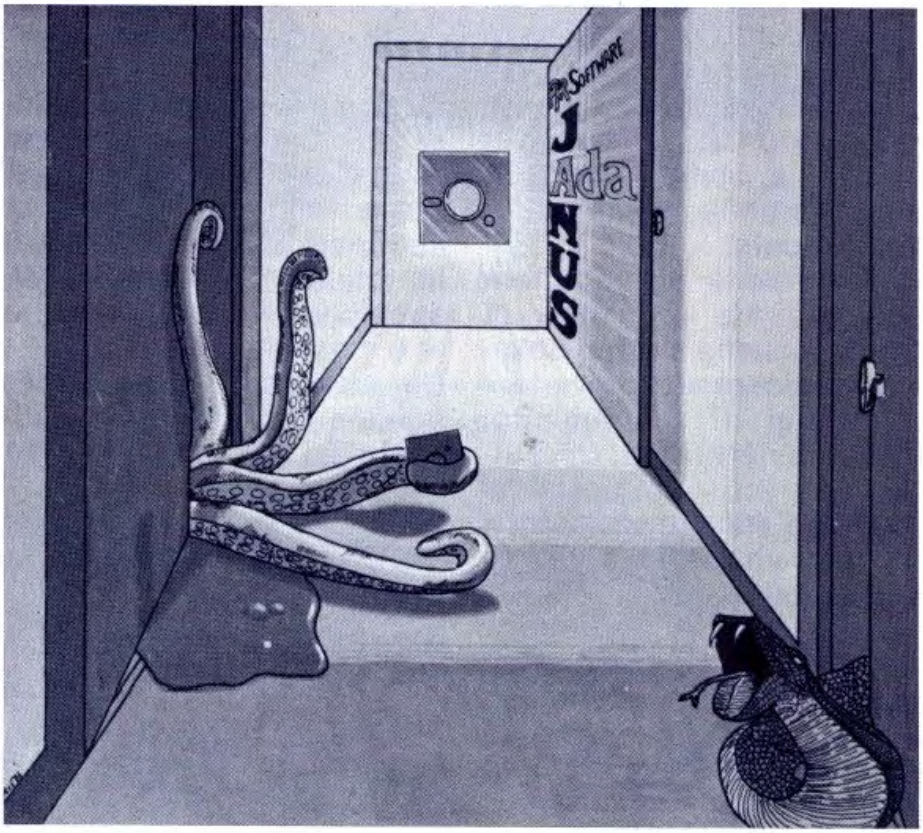
IL GIOCO

All'inizio vi verrà chiesto il livello di difficoltà che volete affrontare, secondo il quale il Computer mescolerà il Cubo Magico. Partite con il primo livello, molto facile, e cercate di risolvere il nono, cosa praticamente molto difficile all'inizio.
Potete sfruttare due tipi di movimenti, spostando separatamente righe o colonne di due facce a scelta. Per far questo ocorre rispondere alle domande nell'ordine:
1) (R) per scelta della riga, oppure (C) per scelta della colonna; segue quindi il numero (1/4) di R o C prescelta.
2) il numero della prima faccia in cui operare lo scambio.
3) il numero della seconda faccia in cui operare lo scambio.
Il TI 99 ricorda quante mosse fate, e ve lo segnala alla fine - sempre che ci arriviate - dopo essersi congratulato.
La risoluzione del gioco può richiedere un tempo che varia tra i pochi minuti del livello 1 alle... ore del livello 9.
Ricordate che il Computer, all'inizio, impiega qualche secondo (o minuto, in proporzione al grado di difficoltà richiesto) per mescolare in maniera adeguata il cubo: il problema, dunque, è *sempre risolvibile.*

(P.V.)

NIBBLER

Questo programma, in EXTENDED BASIC, è la versione per TI 99 del famoso gioco da bar: «Nibbler». Per quei pochi che non lo conoscessero, il gioco consiste nel guidare un serpente che si snoda sullo schermo, chiuso in un labirinto. Ogni volta che il «serpentone» mangia il cibo si allunga di uno o più segmenti, a seconda del livello di difficoltà, che varia, di quadro in quadro. Il Nibbler non può passare su sé stesso, né andare indietro. Riteniamo che un punteggio superiore ai 12.000 punti sia di tutto rilievo (in redazione non abbiamo superato il 6° quadro). Quindi non fatevi ingannare dalla bassa velocità iniziale: dopo il 4° quadro, pregherete che il Nibbler rallenti! Inoltre attenzione: se fate una mossa sbagliata il NIBBLER se ne accorgerà immediatamente, per cui niente false coincidenze!



CARATTERISTICHE DEL PROGRAMMA

170	Dimensionamento matrici
180-240	Definizione caratteri
250	Richiama la presentazione
610-630	Colorazione schermo e 4 Sprites per la scritta Nibbler
770-780	Controllo tastiera o joystick e ritorno al programma principale
270-280	Lettura e stampa labirinto
380	Controllo fine quadro
500	Controllo «GAME OVER»
800	Data dell'intestazione
820	Data labirinto
880	SUB KEYST per il controllo tastiera
890	SUB LABIRINTO: cambia ad ogni schermo, colore di sfondo, colore del serpente, colore e forma del cibo, colore dello schermo.

Dopo il «RUN», se nessun tasto viene premuto, il programma parte da solo. Questo avviene anche dopo aver perso la partita. Buon Divertimento!!!

(M.M.G.)

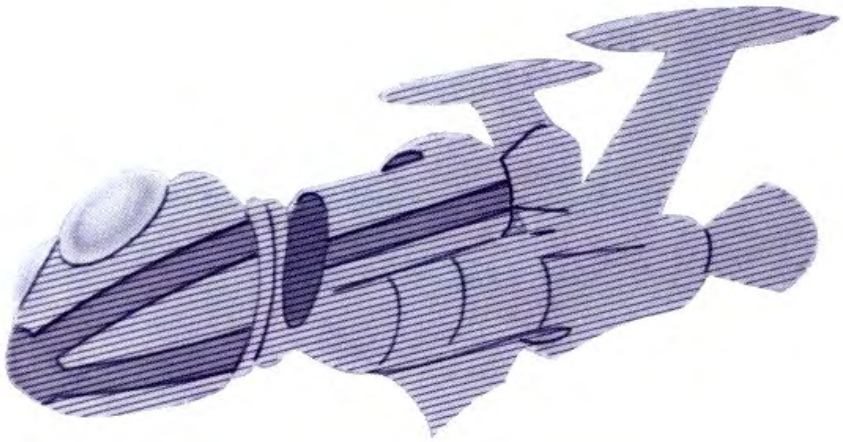
DIFESA

L'obiettivo di questo lavoro è stato cercare di realizzare in extended basic un arcade game veloce, normalmente realizzato in linguaggio macchina, come il PARSEC prodotto dalla stessa Texas Instruments.
Naturalmente per ottenere una discreta efficienza di tutto l'insieme si sono dovute sacrificare alcune varianti (tunnel di rifornimento, sparo da parte degli UFO attaccanti ecc...), ma graficamente e funzionalmente il gioco è abbastanza veloce e stimolante.
Le maggiori difficoltà nella realizzazione di questo tipo di programmi risiedono nella fase di taratura e nella ottimizzazione dei cicli ripetitivi principali, contenenti i test sulle coincidenze, i movimenti della leva di comando e il test sul pulsante di sparo.

IL GIOCO

Come già anticipato il gioco richiede i telecomandi (per cui è indispensabile ricordarsi del tasto ALPHA LOCK) e il modulo EXTENDED BASIAC. È però possibile con una piccola modifica, descritta più avanti, *abilitare la tastiera* al posto dei telecomandi.
Dopo il RUN appare la videata di presentazione con il titolo; questa videata rimane sullo schermo il tempo necessario per la definizione delle forme grafiche e l'inizializzazione di alcune variabili.
A questo punto inizia il gioco. Il «campo di battaglia» è costituito da un cielo, con l'astronave terrestre di difesa sulla sinistra, uno sfondo scorrevole in basso, sotto al quale vengono riportate le astronavi ancora in gioco (5 in totale), il carburante disponibile e i punteggi parziale e massimo ottenuto.
Gli attaccanti sono 5 per ognuna delle 5 forme grafiche; ogni forma ha un grado diverso di pericolosità. Il giocatore deve evitare di farsi investire dall'attaccante, muovendo verticalmente la propria astronave, e contemporaneamente cercare di colpirlo. Se l'attaccante viene colpito esplode e ne entra in gioco un altro, fino all'eliminazione dei 5 che appartengono alla forma grafica in gioco.
Se uno di questi 5 investe l'astronave sarà questa ad

esplodere e il gioco riprenderà dal primo dei 5 attaccanti con la conseguente perdita di una unità.
Ogni cambiamento di forma grafica degli attaccanti viene preavvisata da un suono.
Il carburante di ogni astronave basta per l'abbattimento di tutti i 25 attaccanti, in caso contrario l'astronave esploderà.
Al raggiungimento di 5000 punti viene concesso un bonus di una astronave.



Per abilitare la tastiera vanno modificate le seguenti linee:

```
550  CALL KEY (0,K,S):: CALL MOTION (§1,24*((K = 69)-(K = 88)),0):: IF K = 70 THEN GOSUB 600
1210 CALL MOTION(§2,0,VR):: CALL KEY(0,K,S):: CALL MOTION(§1,24*((K = 69)-(K = 88)),0):: IF CO THEN GO-
     SUB 700 :: GOTO 450
1310 CALL MOTION(§2,VC,VR):: CALL KEY(0,K,S):: CALL MOTION(§1,24*((K = 69)-(K = 88)),0)
```

A questo punto per muovere verticalmente l'astronave si usano i tasti «E» per l'alto,«X» per il basso e «F» per lo sparo.

CARATTERISTICHE DEL PROGRAMMA

LISTATO

Linee	descrizione della funzione
10- 100	Presentazione e set colori
110- 160	Caricamento della tabella (NEM$) delle forme grafiche degli attaccanti
170- 240	Definizione degli 8 caratteri per lo sparo
250- 330	Definizione delle altre forme grafiche
500- 560	Routine principale. Questo, assieme alla routine di sparo, è il cuore del programma, poiché dalla sua efficienza dipendono la velocità di risposta ai movimenti della leva di comando (la numero 2) e del pulsante di sparo. Sono presenti anche due test sulle coincidenze
600- 660	Sparo dell'astronave di difesa. Anche questa routine ha richiesto una taratura per ricercare il miglior tempo di risposta
700- 800	Esplosione dell'astronave di difesa. La rappresentazione grafica dell'esplosione è stata effettuata attraverso 2 forme grafiche alternate per dare l'impressione del movimento.
840- 920	Aggiornamento dei dati riportati in basso sullo schermo e controllo del termine del gioco.
960-1050	Schedulazione della partenza degli attaccanti.
1090-1140	Controllo dell'eventuale record a fine gioco e richiesta di continuare
1180-1230	Partenza del quinto attaccante
1270-1330	Partenza del primo attaccante
1370-1630	Subroutines richiamate ad ogni partenza di nuovo attaccante. Agiscono solo in occasione della partenza del primo della serie di 5 per ogni forma grafica.

VARIABILI

NEM$(1...5) ...contiene le stringhe per la definizione delle forme grafiche degli attaccanti.
ATT .....contatore della forma grafica in gioco.
BO .....switch indicante l'avvenuta assegnazione del bonus.
CO .....variabile per il controllo delle coincidenze
I .....campo di comodo e di controllo loop principale
K .....comodo
KAR .....contiene alternativamente i codici ASCII 100 e 104 per l'esplosione dell'astronave.
NA .....varia da 1 a 5 nell'ambito di una forma grafica di attaccante.
R .....contiene il numero di riga calcolato in base alla posizione dell'astronave al momeno dello sparo.
S .....status del pulsante di sparo.
S1 .....comodo
SCO .....contiene il punteggio accumulato.
TOP .....contiene il punteggio più alto conseguito.
VALO .....valore dell'attaccante (per il punteggio).
VC .....velocità verticale del primo attaccante.
VR .....velocità orizzontale (aumenta progressivamente).
VS .....velocità dello sfondo scorrevole.
X .....comodo
X1 .....coordinata della posizione dell'astronave.
X2 .....coordinata della posizione dell'attaccante.
Y .....controlla il movimento del joystick
Y1 .....coordinata della posizione dell'astronave.
Y2 .....coordinata della posizione dell'attaccante.

(A.S.)

TI 99 NEWSOFT

ANNO 1 - Numero Zero in attesa di registrazione
GIUGNO 1984 — MENSILE
Direttore: ANTONIO C. LOSITO
Coordinatore Editoriale: VINICIO CIVININI
Art Director: BEPPE RE-FRASCHINI — **Illustrazioni:** ROSARIO SPRECHINO
Collaboratori: SERGIO BORSANI, GIANPAOLO BOTTIN, DANIELE CATALFAMO, PAOLO CIVARDI, MAURO MATTIAZZI, EZIO MONTINI, MARCO SQUINTANI, ALBERTO STRAFILE, SEBASTIANO TOMASELLO, PAOLO VENTAFRIDDA

TI 99 NEWSOFT è una pubblicazione della **EDISOFT snc,** via Stefano Jacini 4 - 20121 MI
Tel. 02/807464 - Telex 324284

IN ATTESA DI REGISTRAZIONE AL TRIBUNALE

© COPYRIGHT EDISOFT snc
TUTTI I DIRITTI RISERVATI
Manoscritti e foto originali, anche se non pubblicati, non si restituiscono ed è vietata la riproduzione anche parziale di testi, fotofrafie e listati.
Pubblicità: AV STUDIO
Via Giuseppe Meda, 23 - Tel. 02/8357155

Abbonamenti: l'abbonamento annuo a 11 numeri di *TI 99 NEWSOFT* costa L. 90.000 per l'Italia; L. 150.000 per l'Europa (spedizione aerea); L. 200.000 per tutti gli altri paesi. L'importo, per assegno non trasferibile o per vaglia postale, va inviato a EDISOFT snc, via S. Jacini 4 -20121 Milano.

Fotocomposizione: VIDEOSTENA (MI)
Fotolito: SCANCOLOR (MI)
Stampa: CGB, Albegno di Treviolo (BG)

Concessionaria per la distribuzione: EUROSTAMPA Srl - C.so Vittorio Emanuele 111 - 10123 Torino
Tel. 011/538166-7

L'editore non si assume responsabilità alcuna sul contenuto degli articoli e/o software inviati.
Per gli articoli firmati e/o siglati da collaboratori esterni, la redazione si assume la responsabilità prevista dalle leggi sulla stampa.

Tutti possono collaborare a TI 99 NEWSOFT, sia con articoli che con programmi. Scriveteci o telefonateci in redazione per avere tutti i ragguagli.

SETTEMBRE 1984 L. 10.000

TI 99 NEWSOFT

EDISOFT

ITALIA - IL COMPUTER, ILLUSTRE SCONOSCIUTO

ASSEMBLER - LA MINI MEMORY

BASIC DA ZERO: SAVE & OLD

ANCORA 8 SUPER PROGRAMMI SU CASSETTA

SOMMARIO

- **DA ZERO, A UNO...** pag. 2
- **COMINCIAMO DALL'INIZIO**
  Inizia una rubrica fissa sul TI-BASIC.
  Questa volta parliamo di SAVE e OLD pag. 3-4
- **HARDWARE: IL MODULATORE TV**
  Qualche utile consiglio per migliorare
  l'output del TI 99 sul TV color di casa pag. 5-6
- **SPRITES**
  Per chi ha il modulo SSS Extended
  Basic, proponiamo un metodo molto sofisticato
  per simulare l'istruzione CALL COINC pag. 6-7
- **LA MINI MEMORY**
  Tutto quello che bisogna sapere su
  questo modulo che ci apre le
  porte sul mondo dell'Assembler pag. 7-9
- **IMPARIAMO A COSTRUIRE UN ARCADE GAME: I NEMICI**
  Il primo di una serie di articoli che vi consentiranno
  di diventare dei perfetti "programmatori"
  di videogamex pag. 10-12
- **ITALIA: IL COMPUTER, ILLUSTRE SCONOSCIUTO**
  Riuscirà l'ITALIA a non perdere il "Treno" della
  rivoluzione INFORMATICA? Facciamo il punto
  della situazione pag. 13-14
- **DEDALO...**
  Ovvero, LABIRINTO in 3 dimensioni,
  un gioco che gira in Basic ed Extended.
  Attenzione a non perdervi... pag. 14-15
- **TENNIS**
  Un semplice gioco in Basic per due persone pag. 16-17
- **CRAZY KOALA**
  State attenti ai pesanti massi che cadono dall'alto,
  mangiate più frutta che potete, ma in fretta.
  La piccola Koala vi aspetta pag. 18
- **PAC MAN**
  Una versione del famoso gioco.
  Gira anche in BASIC pag. 18
- **DIAGRAMMI A STRISCE**
  "Un disegno vale più di mille parole"...
  Così comincia l'articolo di Sergio Borsani.
  Fate girare il programma e ne avrete la
  dimostrazione pag. 19-21
- **SLALOM GIGANTE**
  Allenatevi sulle "piste" del TI 99... Quest'inverno
  sarete in grado di fare discese perfette... pag. 22-23
- **ARITMETICA VELOCE**
  Un programma didattico per chi vuole
  tenersi "allenato" in aritmetica pag. 24-25
- **EASY BUDGET**
  Un bel programmino di utilità che, se usato
  bene, vi farà scoprire i punti deboli del
  vostro bilancio familiare pag. 26-27
- **LA POSTA**
  Voi ci scrivete... Noi vi rispondiamo pag. 28
- **VENDO - COMPRO - CAMBIO - SCAMBIO** pag. 29
- **LA VIGNETTA** pag. 30
- **IL QUESTIONARIO**
  Non dimenticate di compilare e rispedirci
  il questionario. Vi invieremo gli adesivi a
  colori di TI 99 - Newsoft pag. 31-32

TI 99 NEWSOFT

ANNO 1 - Numero Uno
SETTEMBRE 1984
MENSILE

Direttore responsabile:
ANTONIO C. LOSITO

Coordinatore Editoriale:
VINICIO CIVININI

Art Director:
BEPPE RE-FRASCHINI

Illustrazioni:
ROSARIO SPRECHINO

Segreteria:
EMANUELA CHIAPELLO

Collaboratori:
SERGIO BORSANI,
GIANPAOLO BOTTIN,
DANIELE CATALFAMO,
PAOLO CIVARDI,
STEFANO DELLI PONTI,
FRANCESCO MACALI,
MAURO MATTIAZZI,
EZIO MONTINI,
LEVIO PEROTTI,
RICCARDO ROSSI,
MARCO SQUINTANI,
ALBERTO STRAFILE,
SEBASTIANO TOMASELLO,
PAOLO VENTAFRIDDA

TI 99 NEWSOFT è una pubblicazione della **EDISOFT snc**, via Stefano Jacini 4 - 20121 MI
Tel. 02/807464 - Telex 324284
REGISTRAZIONE AL TRIBUNALE DI MILANO N. 323 DEL 7.7.84

© COPYRIGHT EDISOFT snc
TUTTI I DIRITTI RISERVATI
Manoscritti e foto originali, anche se non pubblicati, non si restituiscono ed è vietata la riproduzione anche parziale di testi, fotografie e listati.

Pubblicità: AV STUDIO
Via Giuseppe Meda, 23
Tel. 02/8357155

Abbonamenti: l'abbonamento annuo a 11 numeri di *TI 99 NEWSOFT* costa L. 90.000 per l'Italia; L. 150.000 per l'Europa (spedizione aerea); L. 200.000 per tutti gli altri paesi. L'importo, per assegno non trasferibile o per vaglia postale, va inviato a EDISOFT snc, via S. Jacini 4 - 20121 Milano.

Stampa: CGB, Albegno di Treviolo (BG)

Concessionaria per la distribuzione: EUROSTAMPA Srl - C.so Vittorio Emanuele 111 - 10123 Torino - Tel. 011/538166-7

L'editore non si assume responsabilità alcuna sul contenuto degli articoli e/o software inviati. Per gli articoli firmati e/o siglati da collaboratori esterni, la redazione si assume la responsabilità prevista dalle leggi sulla stampa.

Tutti possono collaborare a TI 99 NEWSOFT, sia con articoli che con programmi. Scriveteci o telefonateci in redazione per avere tutti i ragguagli.

DA ZERO A UNO...

Eccoci qua col numero UNO, finalmente!
Con settembre, TI 99 NEWSOFT uscirà regolarmente ogni mese, tutti i mesi.
Per noi è stata una piacevole sorpresa che è bastato il numero ZERO per fare uscire dal dimenticatoio il NOSTRO TI 99/4A.
A decine ci avete telefonato, ci avete scritto entusiasti per la VOSTRA rivista.
Bene! Da questo numero, la Vostra rivista è "più rivista". Ben 32 pagine più un supplemento con tutti i listati dei programmi nella cassetta. Senza contare, poi, la cassetta con 10 programmi in Basic, Extended e Assembler.
Già in questo numero, noterete l'introduzione di alcune rubriche dedicate al Basic all'Extended e all'Assembler. Queste rubriche sono tenute da esperti con una lunga esperienza di programmazione sul TI 99. Ricordatevi, però che i Vostri contributi sono indispensabili. Per questo le Vostre domande saranno preziose per capire meglio il Nostro TI 99. Quindi, continuate a scriverci, a mandarci i vostri programmi che, se pubblicati, vi verranno retribuiti.
Intanto, vi stiamo preparando altre sorprese.
Una di queste, sono gli adesivi a colori di TI 99 NEWSOFT. Un modo per riconoscersi e per dire a chi possiede un'altro computer, che il TI 99 e la sua rivista, esistono!

COMINCIAMO DALL'INIZIO

La prima cosa che dobbiamo imparare per utilizzare un home computer come il TI 99/4 è come utilizzare alcune parole chiave del linguaggio BASIC.
Come tutti, o quasi, sappiamo, ogni computer ha bisogno di una memoria di massa, registratore a cassette o unità a dischi, dove vengono memorizzati i dati dopo essere stati convertiti da impulsi digitali in segnali sonori, nel caso si usi una unità a cassette.
Tale necessità è data dal fatto che il computer non è in grado di mantenere in maniera permanente i dati da noi immessi in memoria.
Infatti quando il computer viene spento, i circuiti delle memorie RAM, dove sono contenuti i nostri dati, non vengono più alimentati, e i dati in esse contenuti si cancellano.
Ci sono dei modi per evitare che ciò avvenga, ma questi metodi obbligano l'utente a intervenire in maniera rischiosa sui circuiti del computer, per i più esperti, descriveremo questa tecnica detta «alimentazione a tampone» più avanti.
Il TI 99/4a dispone di due istruzioni che abilitano il computer a trasmettere dati ad una unità periferica specificata dall'utente, SAVE unità periferica, e a ricevere o caricare dati da una unità periferica specificata, OLD unità periferica.

IL REGISTRATORE A CASSETTE

Il nostro «TI» è predisposto per pilotare 2 unità di questo tipo, di cui una può essere usata indifferentemente per le operazioni di SAVE e OLD, CS1, e una solamente per le operazioni di SAVE, CS2.
Avremo quindi le combinazioni della Fig. 1.

SAVE	OLD
-1- SAVE CS1	-1- OLD CS1
-2- SAVE CS2	-2- non possibile

Figura 1

INGLESE	ITALIANO
SAVE	
REWIND CASSETTE TAPE CS1 THEN PRESS ENTER	RIAVVOLGI IL NASTRO DI CS1 POI PREMI ENTER
PRESS CASSETTE RECORD CS1 THEN PRESS ENTER	PREMI IL TASTO RECORD DI CS1 POI PREMI ENTER
RECORDING	REGISTRAZIONE IN ATTO
PRESS CASSETTE STOP CS1 THEN PRESS ENTER	PREMI IL TASTO STOP DI CS1 POI PREMI ENTER
CHECK TAPE (Y OR N)	CONTROLLI LA REGISTRAZIONE S/N
RISPOSTA -NO- L'OPERAZIONE TERMINA	
RISPOSTA -SI- ALLA DOMANDA	
REWIND CASSETTE TAPE CS1 THEN PRESS ENTER	RIAVVOLGI IL NASTRO DI CS1 POI PREMI ENTER
PRESS CASSETTE PLAY CS1 THEN PRESS ENTER	PREMI IL TASTO PLAY DI CS1 POI PREMI ENTER
CHECKING	CONTROLLO IN ATTO
DATA OK PRESS CASSETTE STOP CS1 THEN PRESS ENTER	DATI OK PREMI IL TASTO STOP DI CS1 POI PREMI ENTER
NEL CASO VI SIANO DEGLI ERRORI DI REGISTRAZIONE	
ERROR - NO DATA FOUND	ERRORE - NESSUN DATO PRESENTE
ERROR DETECTED IN DATA	ERRORE NELLA LETTURA DEI DATI
PRESS R TO RECORD CS1 PRESS C TO CHECK PRESS E TO EXIT	PREMI R PER REGISTRARE SU CS1 PREMI C PER CONTROLLARE I DATI PREMI E PER USCIRE

Figura 2

INGLESE	ITALIANO
O L D	
REWIND CASSETTE TAPE CS1 THEN PRESS ENTER	RIAVVOLGI IL NASTRO DI CS1 POI PREMI ENTER
PRESS CASSETTE PLAY CS1 THEN PRESS ENTER	PREMI IL TASTO PLAY DI CS1 POI PREMI ENTER
READING	LETTURA IN ATTO
PRESS CASSETTE STOP CS1 THEN PRESS ENTER	PREMI IL TASTO STOP DI CS1 POI PREMI ENTER
DATA OK PRESS CASSETTE STOP CS1 THEN PRESS ENTER	DATI OK PREMI IL TASTO STOP DI CS1 POI PREMI ENTER
NEL CASO VI SIANO DEGLI ERRORI DI REGISTRAZIONE O RILETTURA	
ERROR NO DATA FOUND	ERRORE - NESSUN DATO PRESENTE
ERROR DETECTED IN DATA	ERRORE NELLA LETTURA DEI DATI
PRESS R TO READING CS1 PRESS C TO CHECK PRESS E TO EXIT	PREMI R PER LEGGERE SU CS1 PREMI C PER CONTROLLARE I DATI PREMI E PER USCIRE

Figura 3

Una volta attivata una di queste due opzioni, il computer ci fornirà dei messaggi guida che ci aiuteranno nell'esecuzione dell'operazione prescelta.
Nelle fig. 2 e 3 troverete la traduzione completa dei messaggi nelle varie combinazioni (SAVE e OLD).
Purtroppo, non sempre il trasferimento dei dati da computer a registratore e viceversa avviene regolarmente, ed ecco apparire i tanto temuti messaggi di errore.
Esaminiamo il primo messaggio -ERROR no DATA FOUND-.
Questo messaggio appare quando il computer non riceve alcun dato dall'unità specificata, oppure il ricevere con livelli, di volume e di tono, inadeguati.
In pratica nella maggior parte dei casi, per ovviare alla seconda condizione, è sufficiente alzare il volume del registratore, o regolarne diversamente i toni.
Vediamo ora il secondo -ERROR DETECTED IN DATA-.
Questa segnalazione di errore si verifica in uno dei seguenti casi:
— Alterazioni del nastro dovute a cause diverse.
— Livello del volume troppo alto.

EZIO MONTINI

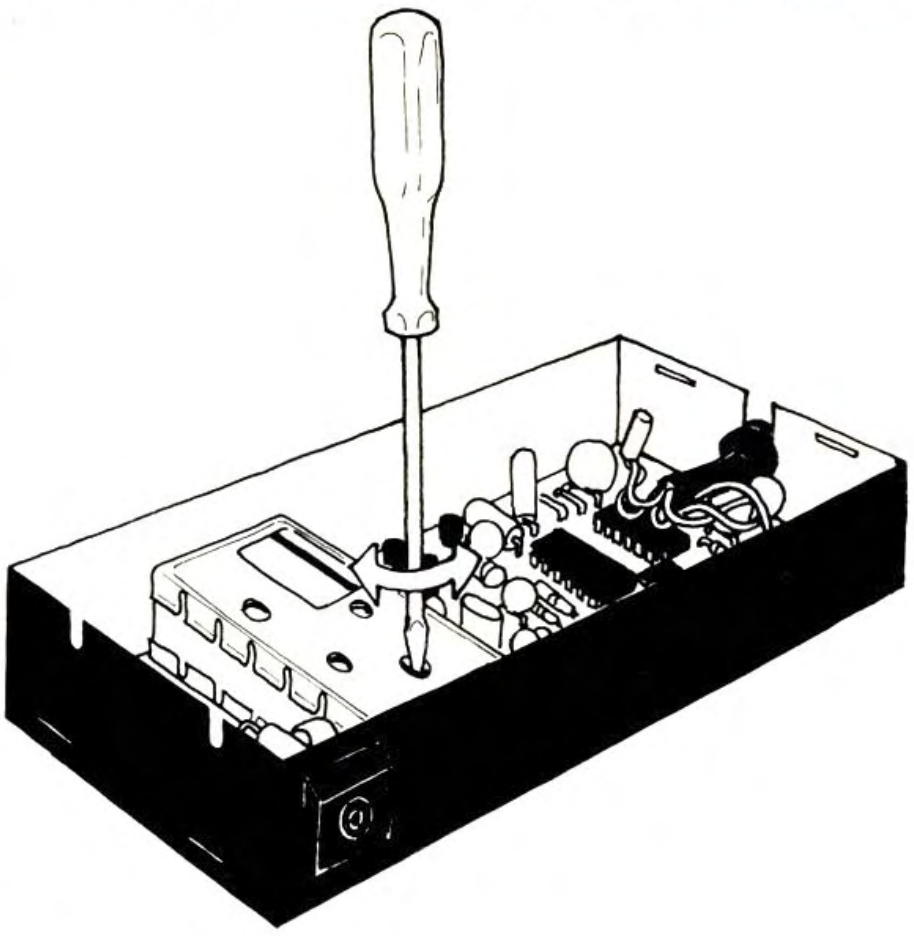
IL MODULATORE TV
UN UTILE CONSIGLIO PER MIGLIORARE L'OUTPUT SUL VIDEO

Come è noto la maggior parte dei piccoli computer possono utilizzare come display sia un monitor che un normale apparecchio televisivo. Il primo offre il vantaggio di dare una immagine molto più nitida e ferma, il secondo rappresenta invece la soluzione più economica per il semplice motivo che quasi ogni famiglia dispone già di almeno un televisore, magari un vecchio bianco e nero. Per offrire una base di confronto tra i due dispositivi, si consideri che su un monitor si possono generalmente inviare 80 caratteri per riga (non è il caso del TI-99/4A che dispone comunque di 28 caratteri per riga) mentre su un normale televisione, oltre le quaranta colonne, si avrebbero grossi problemi di leggibilità.

Un altro inconveniente al quale è esposto un apparecchio televisivo è il verificarsi di eventuali interferenze da parte di emittenti televisive che possono causare disturbi di varia natura come la presenza di righe ed ondeggiamenti sullo schermo. Le presenti considerazioni non possono venir generalizzate in quanto il segnale che esce dal modulatore è sintonizzato sulla frequenza del canale 36 mentre i canali usati dalle emittenti televisive variano da regione a regione.

Molto spesso capita che la trasmissione fatta su un canale possa essere ricevuta più o meno bene anche su altri canali.

Il segnale in questo caso viene chiamato dai radiotecnici armonica del segnale principale, così succede che una trasmissione sul canale 44 interferisca in prossimità del canale 36 provocando i disturbi di cui si parlava e probabilmente il fenomeno è tanto più sensibile nelle grandi città dove maggiore è il numero delle emittenti. A volte toccando con le mani il breve tratto del cavo coassiale che va dal modulatore all'apparecchio televisivo si può influenzare la ricezione di questi segnali migliorando o peggiorando l'immagine sul video ricevendo così l'erronea impressione di qualche cattivo contatto lungo la linea. Quando si verificano inconvenienti di questo tipo è consigliabile per prima cosa provare il collegamento con un altro televisore; se ciò non dovesse dare un sensibile miglioramento è utile sapere che è possibile intervenire sul modulatore del TI 99/4A. Pertanto, prima di decidere drasticamente l'acquisto di



un monitor si può eseguire questa semplice operazione da soli o, per maggiore sicurezza, con l'aiuto di un radiotecnico. Si allentano un poco le due viti, quindi si scalza il coperchio del modulatore facendo attenzione a non toccare i componenti interni per non danneggiarli. Con un piccolo cacciavite si ruota lievemente la vite per il controllo del segnale modulato come è visibile nella figura 1. E' opportuno segnare la posizione iniziale e non effettuare rotazioni indiscriminate.
Ogni volta che s icompie questa piccola correzione al segnale in uscita è necessario sintonizzare nuovamente l'apparecchio televisivo, magari con i tasti della sintonia fine, se ne è provvisto, e valutare se l'immagine ottenuta è priva dei precedenti difetti. Al termine si colloca con delicatezza il coperchio e si riavvita. Nella maggior parte dei casi, con questa semplice operazione, si può ottenere un sensibile miglioramento dell'immagine o la scomparsa dei difetti dovuti ad interferenze.

SERGIO BORSANI

SPRITES

Tutti i possessori del modulo TI EXTENDED BASIC avranno notato che parecchie pagine del manuale di istruzioni sono dedicate agli SPRITES, ma ciò nonostante la tecnica di manipolazione di queste figure non è molto chiara.
Gli SPRITES sono in grado di svolgere determinate funzioni di movimento e animazione senza interferire minimamente con l'esecuzione di un programma, tuttavia si possono controllare per mezzo di apposite istruzioni BASIC.
Dette istruzioni hanno il compito di modificarne la forma, CALL PATTERN, la dimensione, CALL MAGNIFY, la velocità, CALL MOTION, la posizione, CALL LOCATE.
Non mancano le istruzioni per i controlli della posizione, CALL POSITION, della collisione tra due o più figure, CALL COINC, della distanza tra uno SPRITES e un punto del video, oppure tra due figure, CALL DISTANCE.
Il difetto (se così si può definire) più sentito è la poca efficienza dell'istruzione che si occupa di rilevare le coincidenze tra due figure.
Molte tecniche sono state ideate, una di queste consiste nel riempire il programma BASIC dell'istruzione CALL COINC.
Questa tecnica comporta un grosso appesantimento del programma, riducendone la velocità di esecuzione.
Noi vi proponiamo un metodo molto sofisticato che vi permetterà di simulare tale istruzione.
Esso si basa su cicli temporizzatori ottenuti dalle CALL SOUND.

SIMULAZIONE ISTRUZIONE CALL COINC

```
100 CALL CLEAR :: CALL SCREEN(2) :: CALL CHAR(46,
"0000001818") :: CAL SPRITE (#2, 94, 16, 180, 1, 0, 5)

110 FOR N=0 TO 25 :: RANDOMIZE :: CALL PEEK(-31808, Y, X)
:: CALL SPRITE (#3,65+N, 16, Y/2+1, X+1) :: CALL SOUND (-
60, 660, 8)

120 CALL POSITION (#3, Y, X, #2, R, C) :: CALL SPRITE (#1, 46,
16, R, C, (Y-R) *. 49, (Y-C) *. 49) :: CALL SOUND (476, -3, 14)

130 CALL SOUND (120, 110, 6) :: CALL DELSPRITE(#1) ::
CALL PATTERN(#3,35) :: CALL SOUND (100, 220, 6) :: NEXT N ::
GOTO 110
```

Figura 1

IL PROGRAMMA 1 - Figura 1

Questo piccolo programmino di esempio ci dimostra la perfetta efficienza del nostro sistema di simulazione.
Prima di spiegarlo dobbiamo dirvi che la CALL PEEK eseguita alla riga numero 110 serve per ottenere dei numeri casuali, essa è stata utilizzata solamente per comodità di programmazione.
Tutto il programma è basato sulle CALL SOUND delle righe 120 e 130, modificando la durata delle note, si otterrà un anticipo o un ritardo nella coincidenza, con appropriati valori questa routine può essere adattata a qualunque programma. L'esempio sotto riportato vi fornisce anche un utile algoritmo di ricerca basato sulla formula: **(punto di partenza meno punto di arrivo)** *.49, otterrete come risultato una determinata velocità di riga e di colonna che consentirà allo SPRITE di colpirvi inesorabilmente.
La riga 100 oltre che definire il colore dello schermo, genera uno SPRITE "ASCII (94)" che scorrerà nella parte bassa dello schermo con velocità orizzontale di 5.
Le righe seguenti si occupano di fare apparire e in seguito colpire dei bersagli messi casualmente sul video.

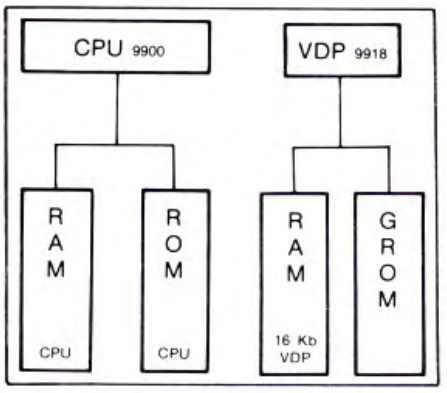
EZIO MONTINI

LA MINI MEMORY

Il modulo SSS Mini Memory aumenta notevolmente le capacità del nostro TI 99, e permette, a chi ha il «coraggio» di addentrarsi nel misterioso mondo dell'Assembler, di sfruttare appieno le sue possibilità, forse incredibili per chi si sia fermato al TI Basic.
La Mini Memory aggiunge in totale 14 Kb di memoria, ripartiti nel modo seguente: 4 K di RAM, 4 K di ROM e 6 K di GROM.
I 16 K RAM(VDP RAM), presenti nella console in configurazione standard, costituiscono il banco di memoria indipendente del Video Display Processor, il microprocessore che si occupa della gestione dello schermo, che ha bisogno di tutti questi bytes nel caso si usi il bit map mode, il modo grafico che permette l'alta risoluzione (eh sì... l'abbiamo anche noi!). Purtroppo normalmente questi vengono utilizzati per la memorizzazione del programma Basic e delle variabili, e così siamo costretti a dimenticarci i 49152 pixel in 16 colori.
I 4 K RAM del modulo sono invece indirizzati direttamente dalla CPU TMS 9900 e per questo motivo in essi è possibile memorizzare programmi in linguaggio macchina. Lo schema riportato, anche se estremamente semplice, chiarirà quanto detto.
Grazie alla presenza di una piccola batteria interna si ha la possibilità di preservare dati e programmi anche a computer spento. Un programma Basic può venire registrato o caricato in un paio di secondi al massimo, con le semplici istruzioni 'SAVE MINIMEM' e 'OLD MINIMEM', con la limitazione dei 4K.
Le ROM e le GROM del modulo incrementano la versatilità del Basic e facilitano la programmazione in Assembler; alcune nuove istruzioni si aggiungono a quelle standard del TI Basic CALL LOAD, CALL PEEK, CALL POKEV, CALL PEEKV, CALL INIT, CALL LINK e CALL CHARPAT.
Vediamo a che cosa servono:
— CALL LOAD(X, Y) è usato per immagazzinare nella locazione X della CPU RAM il valore Y, ed è l'equivalente degli oscuri POKE che troviamo nei listati dei programmi per altri computer. La stessa istruzione può essere usata per caricare da disco programmi in linguaggio macchina: ad esempio
CALL LOAD(«DSK1.RES»)
carica in memoria il file RES dal dischetto nel disk drive I.
— CALL PEEK(X, Y) legge il contenuto della locazione X nella CPU RAM e lo assegna alla variabile Y; equivale al semplice PEEK per altre macchine.
— CALL POKEV e CALL PEEKV hanno la medesima funzione del CALL LOAD e CALL PEEK rispettivamente, ma si riferiscono alla VDP RAM.
— CALL INIT è un sottoprogramma senza parametri che inizializza la CPU memory per subroutines in linguaggio macchina e serve più che altro a chi possiede l'espansione 32K.
— CALL LINK è usato per

passare dall'ambiente Basic ai codici dell'assembly:
CALL LINK(«GO»)
passa il controllo al programma GO; nella Mini Memory c'è una tabella dove è possibile memorizzare i nomi, di 6 lettere al massimo, e gli indirizzi di partenza dei programmi assembly. Ogni assegnazione occupa 8 bytes (6 per il nome+2 per l'indirizzo). La locazione >70IE (dove il segno > indica che il numero che segue è esadecimale) contiene l'indirizzo da cui inizia la ricerca (che arriva fino a >7FFF) del nome e del punto di partenza di un programma in caso di «chiamata» con il CALL LINK. Questo permette anche una lista di parametri, con un massimo di 15 argomenti, per passare dati numerici o stringhe biunivocamente tra il Ti Basic e l'assembly. Se tutto ciò vi sembra complicato, non temete, poichè di questi particolari parleremo più dettagliatamente in uno dei prossimi numeri.
— CALL CHARPAT, infine, permette di assegnare, ad una variabile di stringa, l'identificatore pattern, di 16 cifre esadecimali, di qualunque carattere predefinito e/o creato dall'utente.
Come si è visto, con l'implementazione di queste nuove istruzioni, si rende finalmente possibile il «contatto» diretto con la macchina, un contatto che purtroppo, per vari motivi, il TI Basic (e anche il TI Extended Basic senza espansione di memoria) impedisce. Molti forse non sapranno che farsene di questa possibilità, ma, probabilmente, solo perché non si rendono conto del «passaggio di classe» che essa comporta.

Insieme al modulo viene fornita una cassetta che contiene due programmi: uno, dimostrativo (LINES), ci mostra cosa si può fare ora col TI 99, anche se non in modo del tutto semplice; l'altro è un assemblatore (Assembler Line-by-line).



Esso ci aiuta notevolmente poichè toglie molto del lavoro che è costretto a compiere chi programma direttamente in linguaggio macchina e ha quindi a che fare con numeri apparentemente privi di significato. Con l'Assembler si può infatti programmare con sigle mnemoniche (MOV, LI, BL, DEC,... etc.) senz'altro di più facile apprendimento; ognuna di queste sigle verrà poi immediatamente tradotta dal programma assemblatore in numeri che il microprocessore capisce e divora.
Il programma non è un compilatore ed ha perciò il vantaggio di non avere bisogno dei disk drive, ma, d'altra parte, non è provvisto delle numerose facilitazioni che questo mette a disposizione dell'utente.
Un programma residente su ROM, EASY BUG, colma alcune delle lacune, dando la possibilità di
1) modificare la memoria CPU. 2) visualizzare la memoria GROM. 3) modificare la memoria VDP. 4) mandare in esecuzione programmi assembly. 5) modificare i bits della porta seriale di comunicazione (CRU). 6) registrare e rileggere programmi assembly con il registratore a cassette.
Il modulo si rivela quindi di grandissima utilità, anche se la documentazione, con cui è venduto, è insufficiente per una completa conoscenza delle sue possibilità; se si vuole imparare a programmare in Assembler, è necessario l'acquisto del manuale EDITOR/ASSEMBLER, che dettagliatamente ci descrive le potenzialità della macchina e ci insegna l'uso di questo linguaggio.
Il programma che vi presentiamo in questo numero (di cui vi diamo anche il listato con relative spiegazioni) è un giochino banale, ma riesce a dare un saggio della velocità del linguaggio macchina; richiede l'uso della Mini Memory e dei joystick.
Al listato in Assembler si aggiunge un brevissimo programma Basic che si occupa dei punteggi; esso può essere migliorato, tenendo conto che nella locazione > 7D78 (32120) vi è un valore che determina la velocità di gioco, e nella locazione > 7D96 (32150) è contenuto il numero di caselle «riempite». Lo scopo del gioco è di colorare il più possibile lo schermo, facendo cambiare direzione, per mezzo dei joystick, alla striscia in movimento. Per caricare i due programmi, eseguire le seguenti operazioni: 1) si carichi il DOMINO A con l'istruzione L dell'EASY BUG. 2) si prema il QUIT e si passi al Basic per caricare il DOMINO B.

STEFANO DELLI PONTI

DOMINO

```
7D00 04C3     CLR R3              reset dei registri di posizione (X, Y)
              CLR R4
              CLR R5              reset del registro del punteggio
              LI RO, > 0100       unità di tastiera I (joystick I)
              MOVB RO, δ > 8374
              CLR R14             preparazione dei registri della
                                  direzione
              LI RI5, > 0100
              NOP
          A   BLWP δ > 6020       scansione della tastiera o joystick
              CLR R6
              CLR R8
              MOVB δ > 8376, R6   lettura joystick
              MOVB δ > 8377, R8
              JNE B
              CI R6, 0
              JEQ C
          B   MOV R6, R9          routine di manipolazione dei
                                  dati joystick
              ABS R9
              ABS R10
              C R9, R10
              C R9, R10
              JEQ C
              NEG R6
              SRA R6, 2
              SRA R8, 2
              MOV 6, 14
              MOV 8, 15
          C   SWPB R3
              SWPB R4
              A R14, R4           cambiamento registri di posizione
              A R15, R3
              SWPB R3
              SWPB R4
              CI R3, 31           controllo limiti schermo
              JH ERR
              CI R4, 23
              JH ERR
              MOV R3, R6
              MOV R4, R0
              SLA R0,5
              A R6, R0
              BLWP δ > 602C       controllo ostacoli
              CI RI, > 8000
              JNE ERR
              LI RI, > 8A00
              BLWP δ > 6024       scrittura nuovo carattere
              LI R6, > 2FFF
          D   DEC R6              loop di ritardo
              JNE D
              INC R5              incremento punteggio
              JMP A
              NOP
        ERR   MOV R5, δ > 7D96    scrittura punteggio locazione 7D96
              B * II              ritorno al programma Basic
```

N.B. Se il vostro schermo non visualizza tutte le colonne, aggiungete le seguenti righe al programma Basic

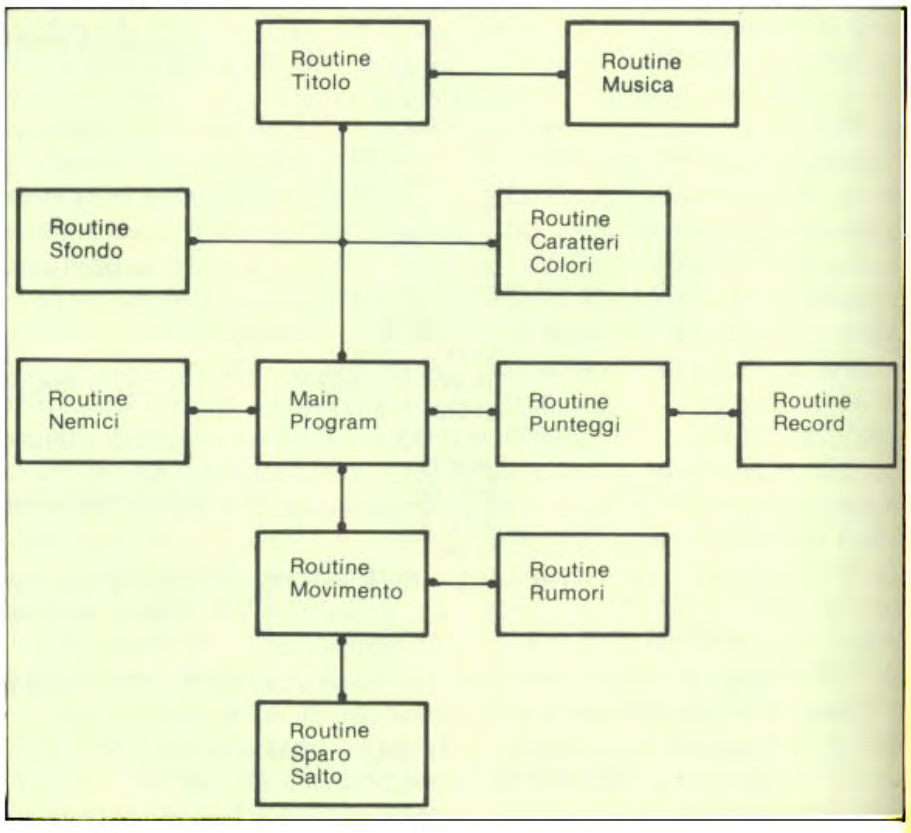
```
135 CALL VCHAR (1, 1, 42, 24)
136 CALL VCHAR (1, 32, 42, 24)
```

IMPARIAMO A COSTRUIRE UN ARCADE GAME: I NEMICI

di Mauro Mattiazzi (MMG)

Costruire un ARCADE-GAME, cioè un gioco d'azione, non è cosa di tutti i giorni, visti i limiti imposti dalle macchine oggi in commercio.
Tuttavia, accontentandosi del compromesso GRAFICA-VELOCITA' è possibile programmare dei giochi soddisfacenti anche in BASIC.
Non è nostra intenzione servirvi dei listati pronti per l'uso a questo penserà la parte di rivista sulla cassetta, ma partendo dal presupposto che il gioco è l'insieme di diverse routines più o meno ripetitive, pubblicheremo di volta in volta delle routines di utilità che serviranno a semplificare il vostro lavoro di programmazione.
Un ultimo consiglio prima di iniziare la programmazione, pardon la lettura: questi consigli sono dettati dall'esperienza, tuttavia non è detto che queste routines non siano migliorabili. Questo anche per rendere il lavoro di copiatura un po' più creativo.
Speriamo che questa serie di articoli stimoli la vostra fantasia ed al più presto ci arrivino dei giochi che sfruttano i nostri consigli o magari ne propongono altri.
Per cominciare inizieremo con lo schematizzare la composizione del tipico ARCADE-GAME: (vedi fig. 1). Tralasciando i passaggi particolari, grosso modo l'insieme del classico gioco spaziale è quello mostrato dalla FLOW-CHART.

Per poter avere un gioco veloce e completo, è importante strutturare il programma impiegando quasi esclusivamente delle subo-routines facendo in modo che il programma le esegua solo se si verificano determinate condizioni, lasciando al corpo del programma vero e proprio il minimo indispensabile del lavoro.
Di volta in volta, analizzeremo una di queste routines mentre a voi spetterà l'applicazione delle suddette nei vostri programmi.
Questo mese parleremo della routine **NEMICI:** parlando di nemici normalmente il pensiero corre a delle scorribande nelle più profonde e lontane galassie.
Ma nemici sono anche i fantasmi del PAC-MAN, oppure le vetture da sorpassare nel POLE POSITION e così via.
Per questo primo articolo prenderemo in esame il primo tipo di nemici, cioè quelli spaziali, con la riserva che grosso modo la generazione e il loro movimento è più o meno uguale a quello degli altri tipi.
Normalmente, la routine dei NEMICI, dipende dai caratteri definiti all'inizio del programma, quindi dopo avere deciso quanti quadri e quanti tipi di nemici devono essere generati nel corso del gioco, bisogna che questa routine amministri le nostre intenzioni.
Il caso più usuale, quello che



si pone principalmente al programmatore è quello delle ondate successive. Ovvero si devono affrontare un certo numero di nemici, quando questi nemici saranno stati distrutti ne verranno generati altri e così via. Questo tipo di routine poi si dividono ancora in due tipi e cioè a seconda se il punto di uscita sul video dei nemici è causale o in un punto ben preciso del video.
Iniziamo ora ad analizzare queste due routines:

```
1° caso
  1000 FOR C=1 TO 5
  1010 K(C)=INT(RND*25)+C
  1020 CALL HCHAR (1, K,
       135)
  1030 NEXT C

2° caso
  1000 DATA 1, 4, 7, 10, 13,
  1010 FOR C=1 TO 5
  1020 READ K(C)
  1030 CALL HCHAR (1, K(C),
       135)
  1040 NEXT C
```

Queste 2 routines sono valide per un'entrata dall'alto dello schermo. Mentre per un movimento laterale, basta cambiare di posto la K e l'uno, ricordandosi di cambiare il valore nella RND onde evitare un BAD VALUE.
Nel caso invece che ci siano diversi pattern per i nemici, è consigliabile inserire i pattern esadecimali in una data, lasciando così liberi i caratteri per altri impieghi.
Per definire poi i nemici, si farà uso sempre dello stesso carattere, man mano che il gioco prosegue.
Questo naturalmente vale se le ondate sono tutte composte dallo stesso carattere, cioè se ogni ondata è composta da un solo tipo di nemico:

```
1000 DATA
     BDE7BD998181824C2,
     3CDB663C4242E7,
     BDE7BD9999413204
1010 CONT=CONT+1
1020 IF CONT=4 THEN 1080
1030 READ K$
1040 CALL CHAR(135,K$)
1050 FOR K=1 TO 13
     STEP 3
1060 CALL HCHAR
     (1, K, 135)
1070 NEXT K
1075 GOTO 1110
1080 RESTORE 1000
1090 CONT=1
1100 GOTO 1020
1110 RETURN
```

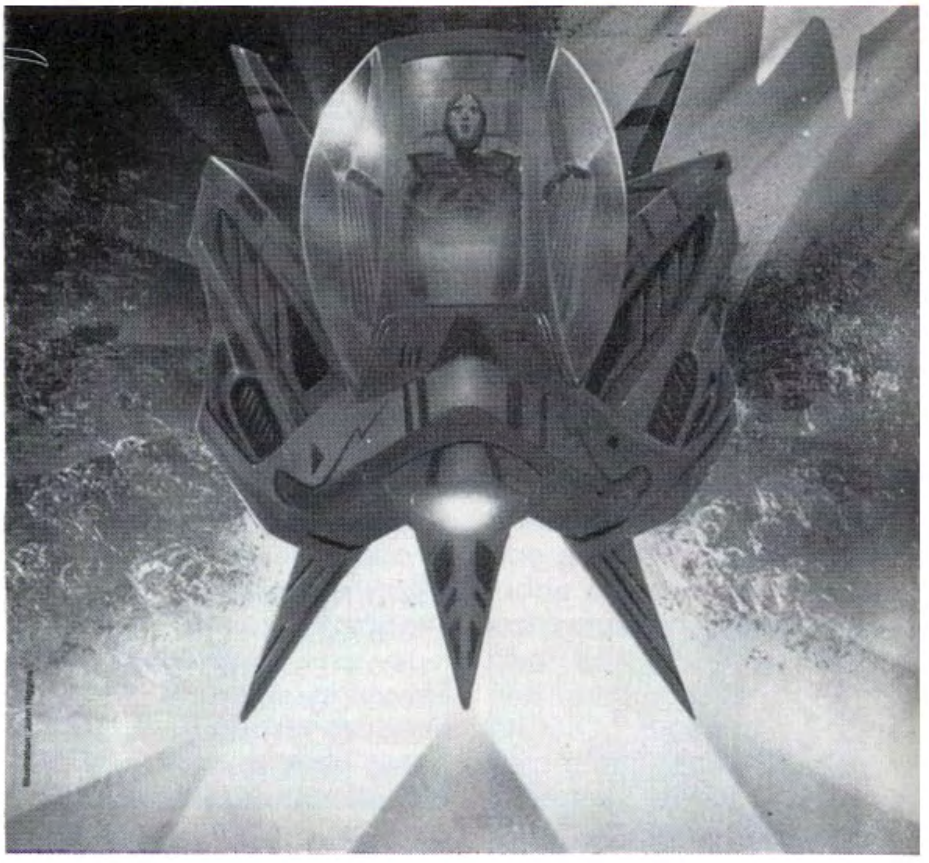
Come abbiamo appena visto usando diversi pattern la routine si allunga notevolmente, tuttavia in questa parte di programma la velocità non è essenziale. La routine sudescritta è valida per il 2° caso, mentre per il primo, è necessario modificare alcune righe nel seguente modo:

```
1050 FOR C=1 TO 5
1060 K(C)=INT(RND*25)+C
1070 CALL HCHAR
     (1, K(C), 135)
1072 NEXT C
```

Queste routines possono sembrare scontate, comunque possiamo garantirvi che la musica cambia quando entra in ballo il movimento dei nemici.
Anche qui la routine deve essere essenziale, per il risparmio di memoria e la velocità, qui elemento essenziale in quanto da essa dipende la velocità del gioco.
Un consiglio che possiamo darvi, per il TI BASIC, è quello di mettere in movimento pochi caratteri.
Primo problema da risolvere nel movimento dei caratteri è il fatto che cancellano quello che incontrano sulla loro strada.
Il problema non sussiste negli sfondi puliti, basta fare stampare in sequenza il carattere rappresentante il nemico, ed un BLANC (carattere 32) nella posizione occupata in precedenza dal nemico.
Riagganciandosi alle routines viste in precedenza, anticipiamo che all'inizio del programma dovranno essere inizializzate alcune matrici, una delle quali è K(n), e l'altra che a noi interessa in questo momento PHS(n, 2): dove **n** rappresenta il numero di caratteri in movimento, **K** rappresenta la coordinata «Y» di partenza, **PHS** infine rappresenta le varie posizioni che opportunamente aggiornate occuperà il carattere da stampare.

```
1° caso
  10 DIM PHS(5, 2), K(5)
  20 FOR C=1 TO 5
  30 PHS(C, 1)=K(C)
  40 PHS(C, 2)=1
  45 NEXT C
  50 REM MOV MOSTRI

  60 FOR C=1 TO 5
  70 P=1
  73 IF PHS(C, 2)
     23 THEN 75 ELSE 76
  75 PHS(C, 2)=1
  76 IF PHS (C, 1)
     30 THEN 77 ELSE 80
  77 PHS(C,1)=K(C)
  80 CALL HCHAR
     (PHS(C, 2), PHS(C, 1),
     32)
  90 CALL HCHAR(PHS(C,
     2)+1, PHS(C, 1)+P, 135)
 100 PHS(C, 2)=PHS(C, 2)+1
 110 PHS(C, 1)=PHS(C, 1)+P
 120 NEXT C
 130 RETURN
```

Questa routine, è valida in ambedue i casi, questo ottimizzando al massimo l'utilizzo di memoria e la velocità di esecuzione.
Naturalmente la routine vista prima serviva per giochi senza sfondo. Se però siamo disposti a sacrificare ancora un poco di velocità, possiamo anche fare muovere dei caratteri su di uno sfondo disegnato senza cancellare niente;
la cosa è ottenibile inizializzando la variabile PHS con un parametro in più.
Ritornando al precedente esempio, il listato viene così modificato:



```
10 DIM PHS(5, 3)
20 FOR C=1 TO 5
30 PHS(C,1)=K(C)
40 PHS(C,2)=1
50 PHS(C,3)=32
60 NEXT C
70 REM MOV MOSTRI
80 FOR C=1 TO 5
81 IF PHS(C,2)
   23 THEN 82 ELSE 83
82 PHS(C,2)=1
83 IF PHS(C,1)
   30 THEN 84 ELSE 90
84 PHS(C,1)=K(C)
86 CALL HCHAR(PHS(C,2),
   PHS(C,1), PHS(C,3))
90 P=1
100 CALL HCHAR(PHS(C,2),
   PHS(C,1), PHS(C,3))
110 CALL GCHAR(PHS(C,2)+
   +1, PHS(C,1)+P,
   PHS(C,3))
120 CALL HCHAR(PHS(C,2)+
   +1, PHS(C,1)+P, 135)
130 PHS(C,2)=PHS(C,2)+1
140 PHS(C,1)=PHS(C,1)+P
150 NEXT C
160 RETURN
```

Ecco che con questa ultima routine possiamo considerare chiuso l'argomento. Per chiudere «in bellezza» alleghiamo al fondo un listato che riunisce ed applica i concetti fin qui svolti.
Un ottimo esercizio sarebbe costruire un gioco utilizzando questo listato; il fondale utilizzato per il listato, è stato preso da un nostro gioco, SPACE PATROL, mentre le routine applicate sono quelle che avete visto nel corso dell'articolo.
Manca quindi solo un cannoncino che spari un poco di rumore un conta punti ed il gioco è fatto.

In ogni caso il listato vale la pena di batterlo, anche solo per vedere la gestione grafica, visto che nel listato vi è anticipato l'argomento della prossima puntata: ROUTINES di GRAFICA e SFONDI FISSI, per quanto riguarda gli SFONDI in MOVIMENTO, ad essi verrà dedicata una puntata intera.

Ed ora un saluto e... sotto a digitare il listato.

(Per i più pigri, il lato A della cassetta contiene in fondo la Routine NEMICI pronta da caricare).

ITALIA: IL COMPUTER, ILLUSTRE SCONOSCIUTO

Nel prossimo secolo la settima potenza industriale del pianeta rischia di passare in serie B

Da un recente studio compiuto dagli americani, risulta che un approccio sistematico al computer deve avvenire entro l'età di dieci anni. E' risaputo che un bambino nel periodo della scolarità elementare compie uno sforzo decisamente inferiore nell'apprendere le lingue, lo studio degli strumenti musicali ed anche nell'uso di questa incredibile macchina. Dicono che i circuiti mentali di un adulto possono adattarsi con gran difficoltà alle problematiche ed alla logica del computer, che cambia giorno per giorno e di cui riesce complesso seguirne l'evoluzione.

Considerazioni che suscitano sconforto per chi scrive, e — immaginiamo — anche per molti lettori, soprattutto in vista della trasformazione del mondo del lavoro, che nel prossimo secolo — diciamo una cinquantina d'anni — vedrà la progressiva ed inevitabile scomparsa dei colletti blu a favore dei «white collars», coloro che opereranno tramite l'elettronica. Elettronica che domani più che mai significa computer, e i tecnici, i manager del prossimo futuro dovranno avere una naturale confidenza con queste macchine.

Gli Stati Uniti ed il Giappone, i due principali colossi mondiali del chip, del circuito stampato e del microprocessore, stanno investendo ingenti mezzi e le migliori energie per dotare i primi gradi di istruzione obbligatoria nei loro rispettivi paesi di piccoli computer, dove i bambini, fin dalla più tenera età, possono prendere visione e studiare le possibilità di applicazione dell'elaboratore.

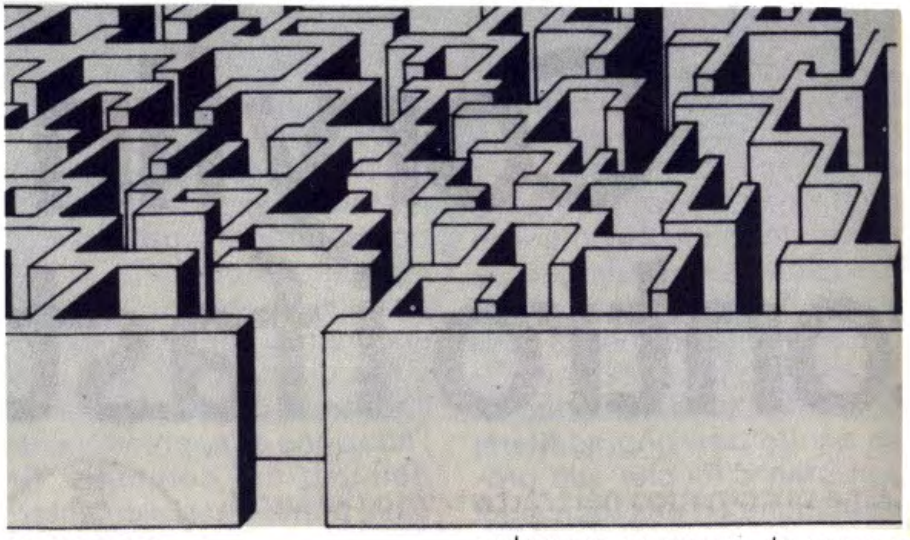
In Italia, invece, non ci si è ancora resi conto dell'importanza di un siffatto tipo d'insegnamento, trascurando l'adozione e l'apprendimento dell'uso del computer fin dalle prime classi elementari. Chiunque sia oggi inserito nel mondo del lavoro nell'Europa occidentale ed abbia una discreta conoscenza delle problematiche di mercato e delle prospettive a livello mondiale, può facilmente comprendere quanto grande sia il divario che separa il nostro continente dagli USA e dal paese del Sol Levante nello studio e nell'applicazione dei computer. Per chi ancora non lo sapesse, i giapponesi stanno investendo milioni di dollari per mettere a punto le macchine della 5ª generazione, quelle con le quali si potrà dialogare senza un codice operativo. Nella Silcon Valley California i migliori ricercatori statunitensi tentano di realizzare una macchina che con prestazioni competitive e le dovute economie di scala sia potenzialmente utilizzabile in ogni nucleo famigliare. Stiamo parlando di grossi sistemi di elaborazione dati, ma anche di piccoli home e personal, che trovano la loro collocazione in qualsiasi azienda e impresa di medio-piccole dimensioni, con la possibilità di gestire una grossa mole di lavoro, data la grande flessibilità del mezzo.

Negli altri paesi della CEE qualcosa sta succedendo. In un congresso tenutosi a Roma lo scorso aprile, dal titolo «Villaggio elettronico: democrazia elettronica», il rappresentante della BBC inglese ha fornito dei dati sull'introduzione del computer in ogni scuola di genere e grado del Regno Unito, nell'ambito di un progetto che dovrebbe fornire macchine alla totalità degli istituti entro la fine del 1978. A supporto di questo sforzo l'ente radiotelevisivo britannico trasmette proprio per le scuole una gran quantità di programmi in software ad uso didattico e, come se non bastasse, la BBC produce anche l'hardware usato dagli studenti.
Analoghe operazioni sono in corso in Germania ed in altri paesi del Nord Europa.
In Italia, se non vogliamo diventare fruitori passivi, è necessario superare il gap tecnologico con rigore e pianificazione, fornire il know-how ai giovani studenti, senza abbandonarci come al solito all'estro e alla creatività latina, che in questo caso, non servirebbero a nulla. Bisogna che gli amministratori della cosa pubblica e la classe politica abbiano ben presente che la diffusione capillare del computer nelle scuole della penisola aiuterà le prossime generazioni a saper usare una macchina che — lo si voglia o no — farà parte della nostra vita quotidiana. In caso contrario perderemmo una scommessa col futuro troppo importante, correndo il rischio di scivolare inesorabilmente nel prossimo secolo fra i paesi del terzo mondo.

FRANCESCO MACALI

DEDALO

Il gioco che vi presentiamo gira anche in TI BASIC.
Questo programma vi permetterà di misurare il vostro senso di orientamento.
Vi troverete infatti immersi in un intricato labirinto a tre dimensioni.



IL GIOCO

Siete rinchiusi in un labirinto; la porta è alle vostre spalle, ma potrete uscire solo dopo aver recuperato il tesoro.
È possibile scegliere fra nove livelli di difficoltà. A ciascun grado corrisponde via via un labirinto di sempre maggiori dimensioni ed un tesoro di crescente valore. La pianta del labirinto viene ricreata ogni volta, per cui dovrete avere un po' di pazienza dato che l'attesa, soprattutto per i livelli di difficoltà superiori, può durare alcuni minuti.
All'inizio del gioco sullo schermo compare la mappa del labirinto; lo stesso accade al ritrovamento del tesoro per consentirvi di ritornare all'uscita. All'occorrenza potrete richiamare la pianta premendo il tasto «H» od il pulsante del joystick, ma ciò vi costerà il dimezzamento del valore del tesoro.
Sulla mappa il simbolo del dollaro «$» rappresenta il tesoro mentre la freccia indica la vostra posizione e la direzione di marcia.
Potete spostarvi all'interno del labirinto sia servendovi del joystick che della tastiera, come sotto illustrato:

TASTO	JOYSTICK	FUNZIONE
E	AVANTI	avanzamento di un passo
S	SINISTRA	rotazione di 90° a sinistra
D	DESTRA	rotazione di 90° a destra
X	INDIETRO	rotazione di 180°

Il punteggio finale tiene conto del livello di difficoltà affrontato, del valore del tesoro riportato, e del numero di passi impiegati per coprire il tragitto.
Buon divertimento!!!

RICCARDO ROSSI

CARATTERISTICHE DEL PROGRAMMA LISTATO

Linee	Descrizione della Funzione
140- 150	dimensionamento tabelle
160- 270	definizione colori
280	prestazione
290- 510	definizione caratteri
520- 630	caricamento tabelle
700- 720	inizializzazione campi
740- 870	selezione difficoltà di gioco
910- 950	posizionamento tesoro
1000-1010	cambio direzione
1020-1060	azzeramento campi per nuova direzione
1020-1060	azzeramento campi per nuova direzione
1070-1160	misurazione lunghezza corridoio
1190-1230	lettura tastiera e joystic
1240-1330	movimento
1340-1420	cambio direzione
1430-1480	urto con il muro
1490-1980	disegna labirinto in prospettiva
1990-2180	disegno cassa del tesoro
2190-2320	disegna aperture a sinistra
2330-2460	disegna aperture a destra
2470-2880	creazione casuale del labirinto
2890-3070	disegna la pianta del labirinto
3080-3350	apertura cassa del tesoro
3360-3490	uscita dal labirinto
3500-3650	stampa punteggi
3660-3700	pausa in attesa che venga premuto un tasto

LISTA DELLE PRINCIPALI VARIABILI

LAB$(21, 29) .....contiene la pianta del labirinto
F$ (3) .....pattern freccia di direzione
D (3, 2) .....tabella di avanzamento
SEC (9) .....tabella tempi di attesa
COL (3, 1) .....tabella colore pareti
ADS$, ADC$, ADP$ .pattern coperchio cassa
LD .....livello difficoltà
CMM, CLL .....dimensione labirinto
P .....distanza dalla parete di fronte
X, Y .....posizione nel labirinto
XT, YT .....posizione tesoro
XP, YP .....posizione parete di fronte
XJ, YJ .....lettura joystic
KK, SS .....lettura pulsante joystic
K, S .....lettura tastiera
DIR .....direzione di avanzamento
Z .....contatore testate
CTP .....contatore passi
U .....flag uscita
T .....flag tesoro
I, F .....contatori
TES .....valore tesoro
PU .....punteggio
PM .....punteggio massimo

Avete acquistato il Numero 1 di TI 99

Ma sapete che è uscito un «Numero zero?»

Per riceverlo direttamente a casa vostra, scriveteci! Ve lo invieremo in contrassegno a L. 10000 (+spese).

EDISOFT
TI 99 NEWSOFT
Via S. Jacini, 4
20121 MILANO

TENNIS

Forse il lettore è a conoscenza dell'enorme difficoltà che si incontra nel realizzare un gioco di movimento in TI BASIC. Con il BASIC esteso si possono utilizzare gli sprite, cioè disegni in movimento, e tutte le istruzioni per il loro controllo.
Tuttavia non tutti posseggono questo modulo di comando e poichè difficoltà non significa impossibilità, è possibile soddisfare le esigenze di chi utilizza il sistema nella configurazione di base.
In questo caso bisogna seguire alcuni accorgimenti di fondamentale importanza. Bisogna limitare al massimo il numero di oggetti in movimento poichè il programma Basic può controllarli solo uno alla volta, sequenzialmente; se gli oggetti fossero, diciamo, 10, oguno di essi dovrebbe aspettare alcuni secondi prima di riavere il proprio turno.
Gli oggetti devono essere piccoli il più possibile, preferibilmente grandi un solo carattere. Poichè il movimento viene creato cancellando l'immagine precedente e ricostruendola in un'altra posizione, più caratteri compongono il disegno, più tempo viene impiegato per eseguire l'operazione e più si nota l'effetto a «mosaico».
Attenzione poi a quando le immagini si sovrappongono. Con gli sprite non ci sarebbe alcun problema, in TI BASIC quando due caratteri si sovrappongono quello per così dire «sottostante» viene cancellato inesorabilmente e dopo il passaggio deve essere ricostruito. Forti di queste prime considerazioni passiamo a considerare il programma. Esso è la simulazione del gioco del tennis e, entro certi limiti, ne rispetta le regole ed i criteri per assegnare i punteggi. Anche il campo di gioco rappresenta abbastanza fedelmente la realtà.
Qualcuno avrebbe potuto desiderare una visione prospettica. Non gli do torto. Il lettore interessato potrà tentare qualche modifica. Le scritte sono realizzate con una subroutine che simula l'istruzione DISPLAY AT, per evitare lo scrolling verticale.
Dopo il titolo appare la richiesta sull'uso dei joystick. L'utente provvisto di tali dispositivi dovrà collegarli alla presa sul lato sinistro della console e, dopo aver risposto «y» (yes) alla domanda, dovrà rilasciare il tasto ALPHA LOCK che blocca le maiuscole.
Se non rispetta questo avvertimento non è garantito l'uso corretto dei joystick.
Le leve consentono il movimento dei tennisti verso l'alto o verso il basso. Vengono ignorati i movimenti laterali o la pressione dei bottoni di fuoco.
Durante il gioco la pallina viene battuta e rimbalza da una parte all'altra del campo. Per ribatterla il giocatore deve intercettarla da fondo campo.
In un primo tempo si era fatto in modo che il rimbalzo della pallina avvenisse solo per contatto con la racchetta, come sarebbe giusto. Tuttavia, per diminuire le difficoltà, nella presente versione, la pallina viene ribattuta anche quando intercetta uno qualsiasi degli altri caratteri che formano i tennisti. Chi è sprovvisto di joystick dovrà naturalmente rispondere «N» alla domanda iniziale.
In questo caso i comandi vengono accettati dalla tastiera. Il giocatore di sinistra dovrà usare i tasti Q e A rispettivamente per il movimento verso l'alto e verso il basso (se non viene premuto alcun tasto il tennista sta fermo).
Il giocatore di destra dovrà usare invece i tasti O e L. Non ci si attende un rispetto rigoroso dello svolgimento della partita, non ci sono né il giudice arbitro né i giudici di linea e non si potrà protestare per un rimbalzo a vostro avviso del tutto regolare.
Il punteggio è automatico e scorre in modo insindacabile. Vince chi raggiunge per primo i tre set. Non sono consentiti né il riposo né il cambio campo.
Se interrompete il programma con la funzione FCTN-4 apparirà il messaggio BREAK IN... con un conseguente scrollo verso l'alto; la partita potrà riprendere con il comando CON, tuttavia lo svolgimento sarà irrimediabilmente alterato.
A proposito, se alzate un po' l'audio del vostro aparecchio TV potrete sentire il rumore della pallina ai rimbalzi.

SERGIO BORSANI

DESCRIZIONE DEL LISTATO

160- 390: definizione dei caratteri e dei colori; il n. 128 rappresenta la pallina, quelli dal 129 al 138 servono per disegnare il campo di gioco, il 124 rappresenta la rete, il 91 e il 92 sono le racchette, i rimanenti i giocatori di tennis.
400- 530: titolo. La più volte richiamata subroutine 2020 simula il comando DISPLAY AT non disponibile in TI basic.
570- 630: richiesta sull'uso dei joystick.
640- 870: disegna il campo da gioco ed i tennisti, prepara le scritte per accogliere i punteggi e pone il valore iniziale ad alcune variabili.
880- 920: ciclo principale che rinvia il controllo alla pallina, al primo giocatore (quello di sinistra), poi ancora alla pallina ed infine al secondo giocatore per poi ricominciare daccapo.
930-1220: controllo del punteggio del primo giocatore.
1230-1490: controllo del punteggio del secondo giocatore.
1500-1580: controllo della battuta e conseguente ripresa del gioco.
1590-1740: messaggio per il vincitore e termine della partita.
1750-2010: subroutine per disegnare il campo da gioco.
2020-2060: subroutine per simulare il comando DISPLAY AT; per essa è necessario specificare le variabili X$, RIGA e COLO.
2070-2090: azzera lo score prima dell'assegnazione di un game.
2100-2140: disegna il tennista n. 2 alla riga specificata.
2150-2170: cancella il tennista n. 2.
2180-2220: disegna il tennista n. 1.
2230-2250: cancella il tennista n. 1.
2260-2450: controlla il movimento della pallina. La linea 2290 fa effettuare il rimbalzo sul campo mentre la linea 2330 rileva l'eventuale rimbalzo sulla racchetta.
2460-2670: subroutine per il movimento del primo tennista.
2680-2890: subroutine per il movimento del secondo tennista.
2900-2970: subroutine per l'assegnazione del punteggio che porta all'assegnazione di un game.

PRINCIPALI VARIABILI

X$: messaggio che deve essere stampato sullo schermo con la routine 2020.
RIGA: riga dello schermo sulla quale deve essere stampato un messaggio o a partire dalla quale deve essere disegnato un tennista.
COLO: come sopra, ma in riferimento alla colonna dello schermo.
JY: variabile flag per indicare al programma se si stanno usando i joystick.
XD: incremento per far avanzare la pallina.
YD: incremento per spostare la pallina verso l'alto o verso il basso.
XP: coordinata orizzontale della pallina.
YP: coordinata verticale della pallina.
R10: numero di riga per determinare la posizione iniziale del primo giocatore.
R20: come sopra, per la posizione iniziale del secondo tennista.
SC1, SC2: punteggi all'interno di un game, i valori vanno da 0 a 3.
PUNTI: variabile per inviare i precedenti valori alla sub-routine 2900 che li trasforma nel classico punteggio 0, 15, 30, 40.
GAME1, GAME2: numero dei game vinti.
SET1, SET2: numero dei set vinti. Con tre set si vince l'incontro.
GG: variabile flag per indicare chi ha la battuta.
CH: contiene il numero di codice ASCII del carattere presente nella locazione dello schermo in cui cade la pallina.
R1, R2: nuove posizioni assunte dai giocatori di tennis.

CRAZY

In questo fantastico videogame impersonate un piccolo ma tenace Koala, che vuole a tutti i costi raggiungere la sua compagna in cima ad una pericolosa impalcatura.
Dall'alto un... Koala rivale(!) cercherà di impedirvi l'ascesa, facendo cadere pesanti massi blu.
A voi il compito di evitarli, cercando anche di cogliere lungo il percorso quanta più frutta possibile.
I guai però non sono finiti: oltre a dover evitare i colpi dell'avversario in alto, ogni tanto un grosso corvaccio giallo interromperà il vostro cammino, tentando di beccarvi sul collo! Non vi rimane in questo caso che... darvela a gambe, salendo o scendendo le scale il più in fretta possibile, o azzardando eroici ma rischiosissimi «salti»!
Non cercate di prendervela comoda: il gioco è a tempo, allo scadere del quale perdete punti e Koala. E' sempre meglio affrettare la marcia verso l'alto piuttosto che soffermarsi a raccogliere frutti.
Per ogni Koala raggiunta guadagnate punti (avete la possibilità di «vincere» una vita) e passate immediatamente ad un nuovo schermo, sempre più veloce e pericoloso.

IL GIOCO

Il movimento del Koala avviene premendo i tasti «E» «S» «D» «X» (MAIUSCOLE: fate attenzione), secondo la direzione delle frecce. Per saltare usate «J», tenendo presente che il salto dura qualche frazione di secondo, durante la quale non siete in grado di correggere il movimento dell'animale (e i massi cadono veloci!).

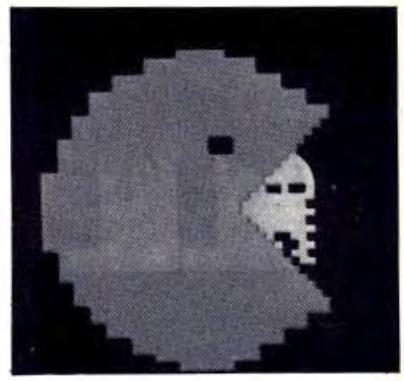
KOALA

CARATTERISTICHE TECNICHE

Crazy Koala è stato scritto in **Extended Basic**, sfruttando al massimo le caratteristiche grafiche di questo linguaggio.
Protagonisti assoluti nel listato sono gli SPRITES, con tutte le istruzioni che li controllano. La grafica sia dello sfondo che delle figure mobili ha reso necessaria la ridefinizione di alcuni caratteri ASCII standard (lettere dell'alfabeto minuscole e maiuscole principalmente). Per «pulire» i caratteri non standard usate un «NEW» prima di lanciare il programma.
E adesso forza, correte dalla vostra Koala!

PAOLO VENTAFRIDDA

PAC-MAN



Questa è una delle tante versioni del PAC-MAN, nato sulla scia del famoso gioco. Ogni parola su questo gioco è inutile; tutto quello che diciamo è che, non essendo certo la velocità una delle caratteristiche principali del TI-BASIC, si è dovuto ricorrere ad alcuni espedienti.
Interessante la memorizzazione del labirinto in una matrice, sotto forma di 0-1 (quindi un codice binario), il che permetta di evitare l'uso delle GCHAR per i controlli e di aumentare notevolmente la velocità di esecuzione.
Chiaramente non si è potuto inserire più di un fantasmino per volta per non rallentare eccessivamente il programma.
Il programma gira in BASIC senza alcuna modifica.

DANIELE CATALFAMO

DIAGRAMMI A STRISCE

Il proverbio (cinese?) «un disegno vale più di mille parole», è uno fra i più citati nella letteratura che tratta la grafica al computer. Oltre al fatto che le immagini grafiche prodotte da un elaboratore sono particolarmente accattivanti tanto da essere spesso usate come veicolo pubblicitario, non va dimenticato che certe rappresentazioni simboliche sono uno strumento di uso comune in molti campi, soprattutto in quello statistico e finanziario. Basta aprire una rivista specializzata o anche un semplice quotidiano per trovare grafici di funzioni, ideogrammi, istogrammi, diagrammi a settori circolari, ecc.

Avete desiderato per un momento un bel diagramma a settori circolari sul video collegato al vostro TI-99/4A? Beh!... in BASIC non credo sia tecnicamente possibile. Ma buoni grafici sotto forma di diagrammi a strisce non costituiscono un problema.

Questi vengono anche detti *diagrammi a colonne o ortogrammi* e sono una delle rappresentazioni più elementari. Il concetto è molto semplice, si tratta di rappresentare i valori contenuti in una tabella con delle strisce di lunghezza proporzionale alla misura alla quale si riferiscono.

Il risultato non contiene certamente più informazioni della tabella, anzi in esso i dati sono generalmente affetti da una maggiore approsimazione, tuttavia offre visivamente, in modo globale, alcuni aspetti riguardanti le relazioni tra i valori riportati che altrimenti non verrebbero colti.

Diagrammi a strisce

1. Inserimento
2. Modifica
3. Tabulazione
4. Ordinamento
5. Diagramma
6. Fine

Fig. 1 - Menu principale. Prima di accedere a qualsiasi sezione del programma è necessario inserire i dati selezionando l'opzione 1.

Questo programma *scritto in TI BASIC* per l'home computer offre il vantaggio di una maggior versatilità rispetto ad altre versioni in quanto consente l'introduzione di dati dipendenti sia da una legge matematica che da una funzione empirica, consente poi la modifica degli stessi in modo selettivo e sequenziale e prevede l'ordinamento dei dati sia secondo l'ordine crescente delle misure sia secondo l'ordine alfabetico degli elementi descrittivi.

L'elaborazione grafica utilizza l'alta risoluzione derivante dalal definizione di caratteri programmabili consentendo la visualizzazione di strisce la cui lunghezza varia in modo discreto pixel per pixel. Il grafico, inoltre, è dimensionato automaticamente e può avere uno sviluppo verticale oppure orizzontale; in entrambi i casi, se il numero di elementi supera il numero di righe e di colonne del display del TI-99/4A, essi vengono mostrati attraverso «finestre» alle quali si passa sequenzialmente premendo un tasto. E' prevista anche la rappresentazione di valori negativi.

C'è di più. Tutto il programma è strutturato in sezioni, selezionabili dal menu principale. Dopo aver introdotto inizialmente i dati, si può accedere ad esse in qualsiasi ordine, potendo ottenere subito il grafico per poi modificare i dati e tornare a rappresentarli magari secondo i valori crescenti.

Dopo il RUN ed il non brevissimo prescan durante il quale il computer riserva aree di memoria alle variabili e cerca eventuali errori nel programma, appare il menu principale. (fig. 1).

Premendo il numero 1 si procede all'introduzione dei dati. Questi possono rispondere o ad una funzione matematica o empirica. (fig. 2).
Nel primo caso si comprende che essi obbediscono ad una formula come potrebbe essere l'interesse composto derivante da un deposito bancario o il decadimento di un elemento radioattivo. Nel secondo caso siamo in presenza di valori ottenuti con una misura sperimentale come potrebbero essere le temperature massime dell'aria osservate di giorno in giorno.

La funzione matematica eventualmente utilizzata va definita alla linea 1000 del programma. Le istruzioni che appaiono sul video sono esaurienti. (fig. 3).

E' da tener presente che, se vogliamo rappresentare ad esempio 20 elementi dipendenti da una funzione matematica, il programma assegna alla variabile indipendente X i valori interi dall'uno al venti. Se si desidera un altro intervallo o un diverso incremento bisogna intervenire unicamente sulla funzione definita alla linea 1000.
Per quanto riguarda gli elementi descrittivi, se si tratta di una funzione empirica, si dovranno digitare da tastiera e potranno essere nomi propri di persona o nomi di monti, di fiumi o i mesi dell'anno ecc.; nel caso delle funzioni matematiche essi vengono posti automaticamente uguali a F(1), F(2), F(3), ecc.
E' possibile intervenire tuttavia o tramite l'opzione modifica o rispondendo affermativamente alla domanda «vuoi numerare i valori F(x) in modo personale?» (fig. 4). Se infatti vogliamo rappresentare l'interesse composto calcolato annualmente, invece di far apparire gli elementi descrittivi F(1), F(2), F(3)... possiamo sostituirli più propriamente con gli anni, ad esempio 1980, 1981, 1982, ecc. (fig. 5).
In questo caso ed ogni qualvolta si desideri la scrittura dell'elemento descrittivo nel diagramma, è necessario ricorrere alla rappresentazione con sviluppo delle strisce in senso orizzontale poichè nei diagrammi a colonne verticali, alla base, c'è il posto solo per numeri progressivi.

Inserimento Dati

Quanti sono i dati? 10 «ENTER»
I dati variano secondo
«1» una legge matematica
«2» una legge empirica

Fig. 2 - E' prevista la rappresentazione di un massimo di 100 dati. Essi possono variare secondo una legge matematica o empirica.

Per inserire la funzione nel programma è necessario interromperlo con il comando FCTN (4) = BREAK e poi scrivere la funzione al numero di linea 1000 usando l'istruzione DEF. Ad esempio:
1000 DEF F(X) = 1000 * (1+.12)^X
scritta la funzione, riprendere l'esecuzione con il comando «RUN».
Se la funzione è già stata scritta, proseguire battendo «SPACE»

Fig. 3 - L'eventuale funzione matematica utilizzata va definita alla linea 1000 del programma.

Vuoi numerare i valori F(X) in modo personale?
(Y/N) «Y»
Scrivi il numero iniziale e l'incremento separati da una virgola.
(N,I) 1975,1 «ENTER»

Fig. 4 - E' possibile numerare in modo personale i valori dipendenti da una funzione matematica. Per default vengono assegnati numeri progressivi.

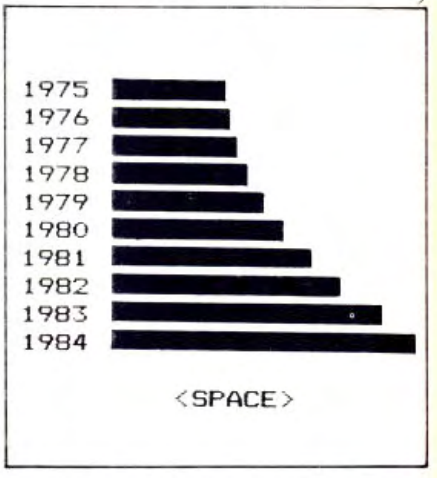


Fig. 5 - Una particolare routine garantisce che la lunghezza delle strisce possa variare pixel per pixel, consentendo un'alta risoluzione grafica.

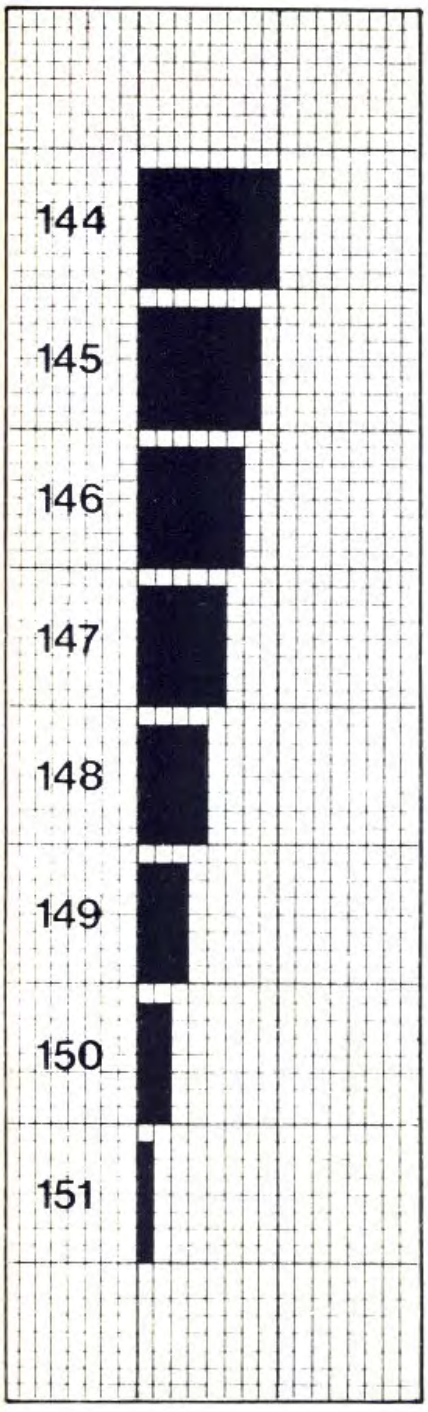


Fig. 6 - Le strisce vengono realizzate stampando di seguito il carattere 144 e completando con il più opportuno degli altri caratteri numerati dal 145 al 151.

DESCRIZIONE DEL PROGRAMMA

Linee	Descrizione Funzione
120- 440:	Dimensionamento delle matrici (linea 120) e definizione caratteri grafici.
450- 640:	Menu principale. Le variabili FLAG e FLAG2 controllano l'accesso ad alcune parti del programma provenendo da diverse sezioni. Il programma è strutturato in modo modulare e le varie sezioni iniziano ai numeri di linea presenti nell'istruzione 640 ON-GOTO.
650- 810:	Menu per l'inserimento dei dati.
820- 990:	Istruzioni per definire la funzione matematica nel programma.
1000-1030:	**Funzione matematica per il calcolo del capitale con interesse composto. Essa, naturalmente, può essere cambiata dall'utente.**
1040-1130:	Nel caso di una funzione matematica, se gli elementi descrittivi dei dati da rappresentare non vengono personalizzati dall'utente, sono posti automaticamente uguali a F(1), F(2), F(3), ecc. Le variabili MIN e MAX memorizzano erispettivamente il minimo ed il massimo della funzione per il dimensionamento automatico.
1140-1270:	Inserimento dei dati nel caso di una funzione empirica. A$(J, 1) è l'elemento descrittivo mentre A$(J, 2) è la misura corrispondente. Il test della variabile INS indica in ogni momento l'avvenuto inserimento dei dati (1260).
1280-1380:	Sotto-menu per la modifica dei dati.
1390-1550:	Modifica sequenziale. Alle linee 1500-1530 si verifica se devono essere variati i valori di MIN e MAX.
1560-1770:	Modifica selettiva. In questo caso si modificano solo i dati richiamati con il loro numero d'ordine.
1780-2040:	Tabulazione dei dati. La variabile flag SCREEN non ha nulla a che vedere con l'istruzione CALL SCREEN; essa ha il solo scopo di indicare quando sullo schermo sono presenti 20 righe di dati. Infatti, ogni venti righe lo scrolling si arresta per permettere la lettura e riprende alla pressione di SPACE.
2050-2100:	Breve routine per stampare il messaggio: «PREMI SPACE».
2120-2220:	Sotto-menu per l'ordinamento. Gli elementi di tabella possono venir ordinati in ordine alfabetico o secondo le misure crescenti.
2230-2300:	L'ordinamento è ottenuto richiamando la subroutine 2340. Per la stampa si utilizza la precedente sezione di programma dopo aver posto uguale ad uno la variabile FLAG.
2310-2590:	Subroutine per l'ordinamento. Poichè anche le misure sono memorizzate in una matrice di stringa e la routine di ordinamento prevede il confronto tra stringhe, vengono utilizzati alcuni espedienti per evitare che, ad esempio, la «stringa» 90 risulti maggiore di 1200, poichè inizia con un codice ASCII maggiore. Altri metodi potevano consistere nel ridurre tutte le stringhe con le misure alla stessa lunghezza, aggiungendo space a sinistra, o nell'usare due routine, una per le misure (numeri) ed una per le descrizioni (stringhe).
2600-2710:	Sotto-menu per tracciare il diagramma. Si può ottenere il diagramma delle misure ordinate o nell'ordine in cui sono state inserite.
2720:	La variabile FLAG1 indica se si devono rappresentare misure ordinate oppure no.
2730-2742:	E' richiamata la routine di ordinamento.
2745-2747:	Se il diagramma non deve rappresentare misure ordinate, vengono attribuiti numeri progressivi al vettore (X(J); in questo modo i dati verranno richiamati secondo l'ordine di inserimento, usando X(J) come indice.
2750:	Se il valore minimo delle misure è negativo si passa alla stampa di un diagramma con strisce verticali dimensionato per la rappresentazione di misure negative, in cui alcune strisce possono essere rivolte verso il basso.
2760-2840:	Sotto menu per la scelta di un diagramma a strisce orizzontali o verticali.
2850-3170:	Diagramma a sviluppo verticale. Se i dati superano il numero di colonne disponibili, essi vengono mostrati attraverso «finestre». Nella scheda tecnica è spiegato come ottenere la lunghezza delle strisce in modo da ottenere l'alta risoluzione.
3180-3430:	Diagramma a strisce orizzontali. In questo caso è possibile stampare, in corrispondenza di ogni striscia, la descrizione.
3440-3830:	Diagramma con valori anche negativi.
3840-4000:	Stampa dei numeri progressivi minimo e massimo dei dati rappresentati in una «finestra».
4010-4100:	Manda il messaggio «SPACE» usando l'istruzione CALL HCHAR, simulando il comando DISPLAY AT.
4110-4330:	Questa subroutine permette di numerare in modo personale i valori di una funzione matematica. Ad esempio, gli interessi composti maturati potranno esser associati agli anni 1980, 1981, 1982, ecc.
4340-4440:	Subroutine per il controllo dell'avvenuto inserimento dei dati. Ad essa si accede prima di ogni eventuale modifica, prima della tabulazione, dell'ordinamento e della stampa del diagramma.
4450-4490:	Fine del programma.

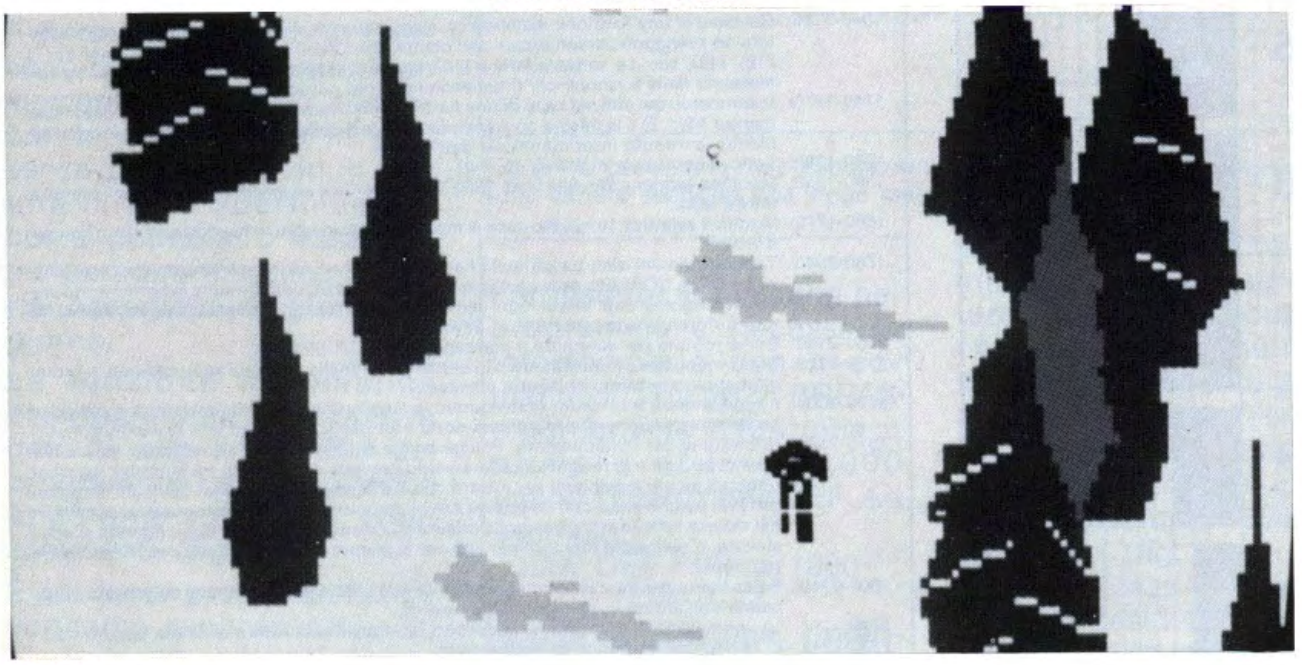
SCHEDA TECNICA

Per far variare le colonne del grafico di quantità discrete di 1 pixel o punto video si è ricorsi ai caratteri programmabili con la funzione CALL CHAR.
Ogni carattere occupa una piccola area formata da 8x8 punti i quali possono essere colorati indipendentemente con un colore loro proprio o lasciati con il colore dello sfondo. Si sono pertanto definiti caratteri «interi», lunghi 8 pixel e caratteri «frazionari» lunghi 7, 6, 5, 4, 3, 2, 1 pixel.
Quando si stampa una striscia del grafico prima si calcola un certo numero di caratteri «interi» e poi si completa con un opportuno carattere «frazionario» in modo che la lunghezza complessiva sia quella desiderata.

SERGIO BORSANI

SLALOM GIGANTE

Siete dei buoni sciatori?
Se si questo gioco vi terrà in allenamento per il prossimo inverno, altrimenti sarà una buona occasione per imparare divertendosi.



IL GIOCO

Questo gioco ci propone, 23 livelli di difficoltà e ben 20 tipi diversi di tracciato.
Non fatevi illudere dal titolo, pensando che questo sia il solito e noioso gioco del «chi schiva più bandierine vince», al posto delle bandierine troverete ad aspettarvi gli ostacoli più disparati, case, tratti di pista pericolosamente ghiacciati, alberi, e tanti altri; a voi il compito di scoprirli.
Il nostro compito è di condurre questo spericolato sciatore attraverso i tracciati che il computer ci proporrà di volta in volta.
Potremo guidarlo con i tasti - S - (sinistra) - D - (destra), oppure azionando la leva del JOYSTICK 1, dopo aver fatto le opportune modifiche descritte nell'accluso riquadro «DESCRIZIONE DEL PROGRAMMA».
Non cercate di elaborare l'ormai nota tecnica «impara la mossa perché il percorso è sempre uguale».
Ci sono ben 150 possibili combinazioni di percorso e vi assicuriamo che in redazione non siamo mai riusciti a vedere un percorso uguale, anche dopo svariati tentativi.
Rimboccatevi le maniche, mettete la sciolina sugli sci e via.
Non investite troppe persone, case o alberi, potrebbero arrabbiarsi.

PER I POSSESSORI DI UNA STAMPANTE

Desideriamo rendervi noto che il presente programma non è interamente listabile tramite RS232 con la stampante TEXAS, perché nelle ultime 20-30 righe di «DATA» sono presenti dei caratteri di controllo, non visibili listando normalmente su video il programma, che provocano un effetto quanto mai dispendioso e spiacevole - FF «From Feed» - inutile spiegare ciò che significa, i possessori di tale periferica conoscono molto bene il risultato di questo comando.

EZIO MONTINI

MODIFICHE PER ABILITARE IL JOYSTICK n. 1

```
460 CALL JOYST (1, J, O)
461 IF J = -4 THEN 861
462 IF J = 4 THEN 863

861 J = 2
862 GOTO 470
863 J = 3
864 GOTO 470
```

DESCRIZIONE DEL PROGRAMMA - SLALOM GIGANTE -

Linee	Descrizione della Funzione
100- 400	inizializzazione delle variabili, caricamento dati grafici
410- 580	spostamento sciatore e verifica di collisione
590- 600	scelta RND della sequenza del paesaggio
610- 640	visualizzazione commenti di fine gioco
650- 750	richiesta di nuova partita e livello di difficoltà
760- 860	routine per il suono della collisione e visualizzazione incidente
870-1280	dati numerici e di stringa

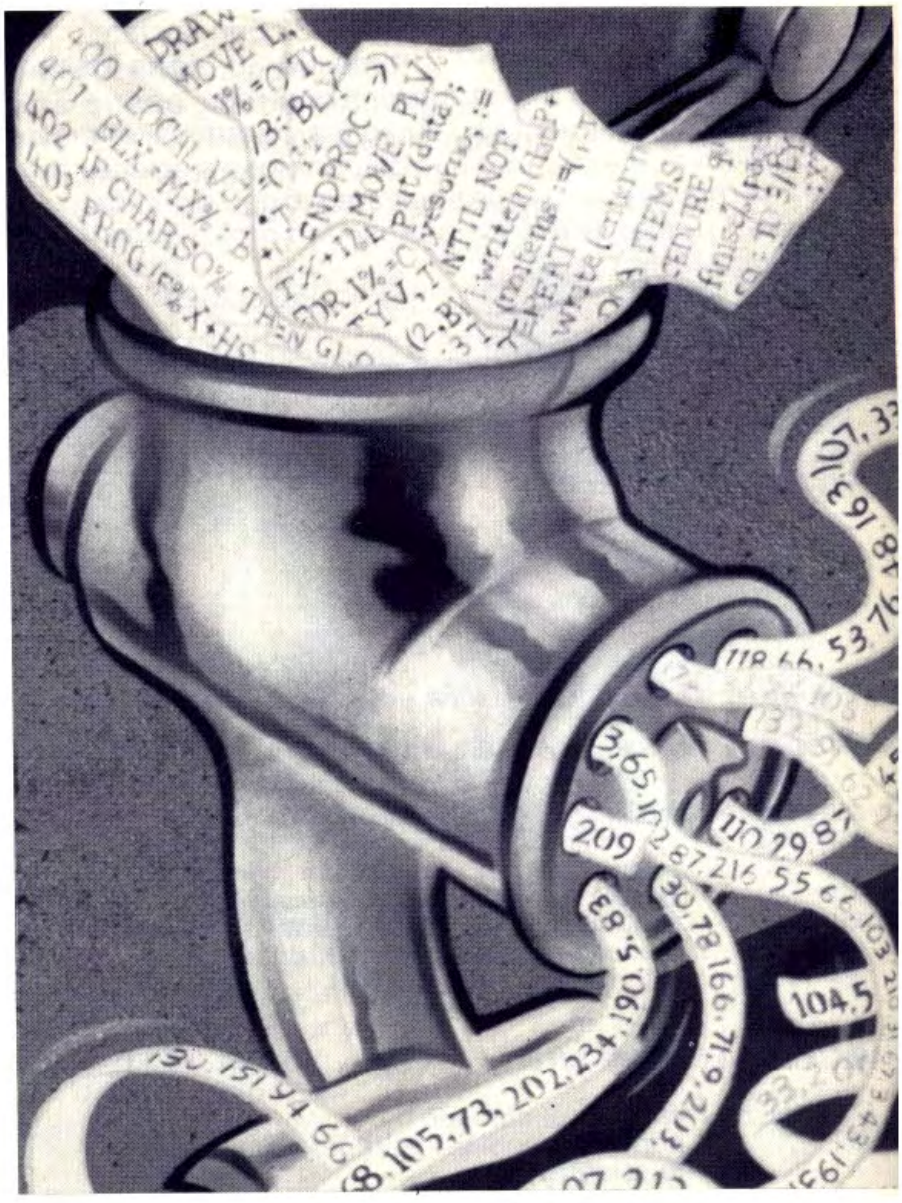
VARIABILI NUMERICHE

A, B, H, K, T variabili di comodo
D .....contiene i codici ASCII per visualizzare lo sciatore nelle varie posizioni
D1 .....contiene ASCII 32 per cancellare la scia
J .....variabile di ritorno della CALL KEY
M .....contatore dei metri percorsi
O .....variabile di stato della CALL KEY
P .....contiene il tipo di percorso scelto
PE .....contatore delle vite rimaste
X .....variabile di ritorno della CALL GCHAR in caso di collisione
Y .....contiene la colonna dove deve essere visualizzato lo sciatore
Y1 .....variabile contenente la colonna per il cancellamento della scia

VARIABILI DI STRINGA

A$ .....stringa contenente le parti di paesaggio da visualizzare

ARITMETICA VELOCE



Le applicazioni del computer alla didattica sono vecchie quanto il computer stesso. Con l'entrata in scena dei microcomputer e la sempre maggior diffusione dei «personal» o degli «home» si sta assistendo ad una inversione di quelli che sono i concetti che stanno alla base e permeano tutta l'attività del settore. Volendo semplificare e dare quasi uno slogan si può dire che si è passati dal grosso computer che programma l'allievo al piccolo computer che viene programmato dall'allievo stesso. Il cambiamento non è stato determinato dall'evoluzione del computer ma rientra nella logica dei nuovi orientamenti pedagogici.

Non è più la macchina che interroga il piccolo utente per verificare se egli ha capito le spiegazioni ma è l'allievo che impara valutando le reazioni e le risposte della macchina a suoi particolari quesiti.

In questa ottica non è necessaria neppure la presenza di un programma. Il computer si può usare in modo immediato oppure saranno gli allievi a creare piccoli programmi che verranno ideati prima a tavolino e poi tradotti in un linguaggio e da essi stessi digitati e verificati. L'esperienza quotidiana mi suggerisce che è questa una delle prospettive più efficaci e di più facile realizzazione.

Un altro modo, certamente più riduttivo, consiste nell'usare il computer come semplice supporto, in sostituzione al materiale cartaceo.

Mi è capitato, sia pur raramente, di scrivere qualche programma «usa-e-getta» in sostituzione a schede didattiche per le esercitazioni di aritmetica usate da mia figlia che frequenta la seconda elementare.

Ritengo che il calcolo veloce non sia elemento trascurabile nella formazione del bambino e contribuisca a fargli affrontare problemi di natura matematica con maggiore sicurezza, anche se probabilmente non c'è una stretta connessione tra lo sviluppo del ragionamento logico e le capacità di eseguire velocemente i calcoli.

In questo caso proprio per la ripetitività degli esercizi il computer può sostituire l'uso di una grande quantità di schede con il vantaggio che è in grado di rilevare subito eventuali errori, annotare il numero delle risposte esatte e fornire il tempo impiegato.

In seguito, un programma creato per questo scopo può essere modificato per cam-

biare il tipo di operazioni, il livello di difficoltà o il modo con cui viene stabilito un eventuale punteggio.
Il programma qui presentato rientra proprio in quest'ottica ed il suo scopo è quello di esercitare il bambino nel calcolo orale.
Per utilizzarlo è necessario semplicemente scegliere uno dei 5 livelli di difficoltà previsti e rispondere alle 10 domande dopo aver eseguito mentalmente le operazioni (addizione, sottrazione, moltiplicazione e divisione). Al termine di ogni domanda si ottiene la risposta e, nel caso di errore, il risultato corretto. Finito l'esercizio, appare il tempo totale impiegato, il numero di risposte esatte ed il punteggio che tiene conto di entrambi gli elementi.

DESCRIZIONE

Una delle difficoltà che si sono dovute superare è la mancanza nel TI-99/4A della funzione di orologio TIMES per cui il tempo dev'essere desunto approssimativamente con l'incremento di una variabile (TT=TT+1). Il problema è che se le risposte venissero accettate con un'istruzione INPUT o ACCEPT AT, abbinate eventualmente ad un'istruzione CALL KEY, appena si iniziasse a digitare la risposta, non verrebbe più incrementata la variabile per il controllo del tempo. In tal modo si potrebbe digitare subito un numero qualsiasi per fermare il tempo e poi pensare a scrivere con tutto comodo la risposta.
Nel programma è stata creata una routine con funzione di data-entry, senza l'uso dell'istruzione INPUT, pur mantenendo le stesse funzioni di editor, come lo spostamento del cursore a sinistra (FCTN S) e la sostituzione di un carattere.
In questo modo il tempo scorre e viene incrementata la variabile TT fino a che non si preme ENTER. In più, prima di premere ENTER, è possibile annullare completamente una risposta premendo il tasto Q (semplicemente, senza il tasto di funzione) per poi riscriverla correttamente (si spera!).

SERGIO BORSANI

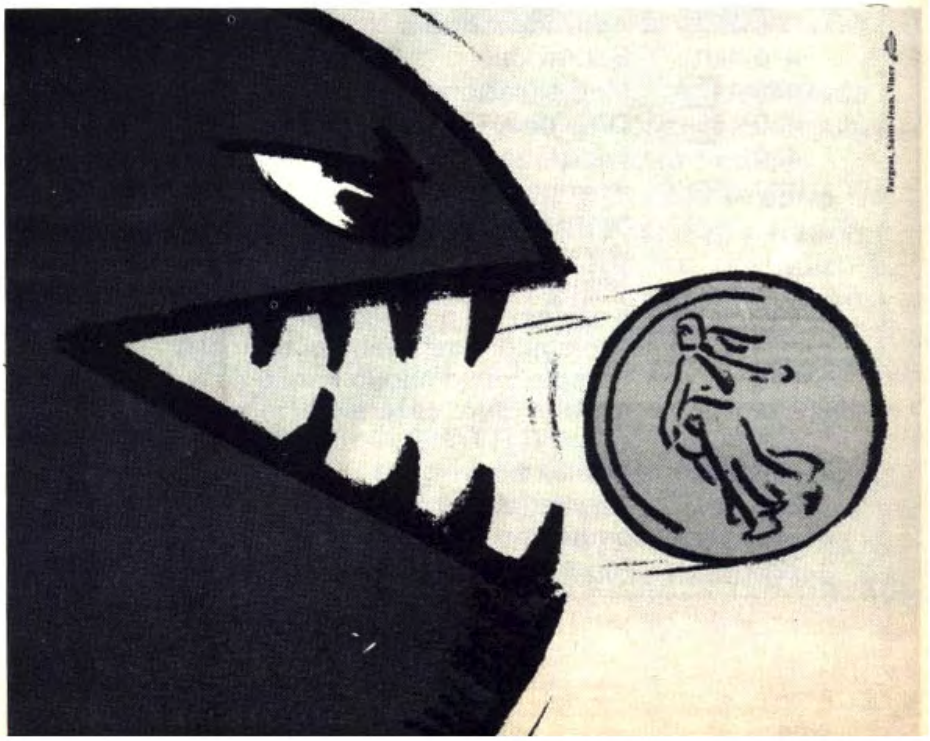
DESCRIZIONE DEL PROGRAMMA

Linee	**Descrizione della funzione**
100-180:	Titolo. Le linee 150-180 creano 28 sprite multicolori con la forma di numeri a dimensioni doppie rispetto ai normali caratteri. I numeri così creati si muovono sullo schermo in tutte le direzioni.
190-290:	Musica che precede l'esecuzione del programma.
300-330:	Scelta del livello di gioco.
340-350:	Inizio e azzeramento delle variabili. TT viene usata per il tempo, ESA per il calcolo delle risposte esatte.
360:	Apertura del ciclo per la formulazione di 10 domande e scelta del tipo di operazione (OP).
370-380:	Rinvio a quattro differenti sezioni di programma corrispondenti alle quattro operazioni. Modificando la linea 380 si può ottenere un ciclo di domande riguardanti sempre la stessa operazione.
390-400:	Addizione. N1 ed N2 sono i numeri da addizionare, RIS è la risposta.
410-420:	Sottrazione.
430-440:	Moltiplicazione.
450-460:	Divisione.
470:	Inizializzazione variabili per la routine con valore di data-entry.
480-550:	Routine con valore di data-entry che siumla l'istruzione ACCEPT AT. A differenza di quest'ultima, durante la digitazione del risultato viene incrementata la variabile TT per il controllo del tempo.
560-580:	Risposte alle domande e chiusura del ciclo.
590:	Il calcolo del punteggio tiene conto del numero di risposte esatte, del tempo impiegato e del livello di difficoltà.
600:	Calcolo del tempo in minuti e secondi. La misura, ovviamente, è approssimata e può essere corretta cambiando il valor 25 nell'espressione TT=INT(TT/25).
610-670:	Musica finale.
680-710:	Risultati consistenti nel tempo impiegato, nel numero di risposte esatte e nel punteggio.
720-750:	Ripresa o termine della prova.

EASY BUDGET

Il titolo inglese di questo programma non deve trarre in inganno. Il «prodotto» è tutto italiano, anzi italianissimo, e significa esattamente «facile bilancio familiare».
Chissà quante volte, sommersi da una valanga di conti, fatture, rate e estratti conto, vi sarà venuta la curiosità di sapere quanto spendete per il vitto, la casa, l'auto o i vostri passatempi. Ora non avrete più la scusa della mancanza di tempo e con un po' di buona volontà potrete facilmente inquadrare la vostra situazione finanziaria, resa ancora più chiara da un istogramma finale a colori.
Una volta lanciato il programma, si presenterà ai vostri occhi su sfondo giallo una simpatica scritta in corsivo, rosso-verde, lampeggiante, accompagnata dallo scorrere di un conto immaginario con i suoi addendi ben incolonnati da sinistra a destra secondo l'ordine matematico delle centinaia, decine, unità. Tale effetto si ottiene simulando all'interno del programma un'istruzione di PRINT USING ottenute alle linee 560-570. Quando lo schermo da giallo diverrà grigio, allora il computer attenderà pazientemente che battiate un qualsiasi tasto. A operazione avvenuta si entrerà nella fase più delicata e importante, quella dell'introduzione dei dati. Come prima cosa vi sarà chiesto dal computer l'ammontare totale delle spese da voi sostenute, quindi si passerà all'impostazione delle singole spese suddivise in nove voci: *tasse, affitto, auto, gas-luce, acqua-telefono, vestiario, alimenti, assicurazione, hobby-viaggi.* Per ogni INPUT verrà subito calcolata e visualizzata in percentuale l'incidenza che quel determinato dato possiede nel contesto delle spese totali. La fase successiva mostrerà inoltre ogni singola voce, identificandola con un numero crescente da zero a nove, e la sua percentuale posta alla destra. Dopo la solita CALL KEY d'attesa si passerà alla tabulazione dell'istogramma. Anche in questa schermata appariranno i valori in percentuale posti vicino alle prime tre lettere che compongono il nome completo della spesa; per esempio, la voce tasse apparirà come TAS, questo per ovvie ragioni di spazio. L'istogramma, utilissimo per dare una visione d'insieme ma nello stesso tempo dettagliata delle spese, sarà a blocchi verticali di colore giallo, blu, rosso, verde. L'ultimo blocco mostrerà tutte le altre spese non comprese nelle nove voci principali. Attenzione a dare sempre dei valori reali nell'immissione dei dati, pena la non esecuzione dalla linea 1530, che controlla il diagramma con conseguente interruzione dello stesso. Vi viene dunque offerta una valida alternativa al solito blocco di carta e matita, sappiatene fare buon uso.

MARCO SQUINTANI



REMARKS

300- 340	Assegnazione colori
350- 370	Assegnazione alcune variabili
380- 500	Assegnazione caratteri
510- 530	Lettura cifre per il conto della prima pagina video
540- 630	Simulazione PRINT USING e rifiniture stampa conto
640- 650	Stampa totale conto
660- 730	Stampa scritta EASY BUDGET in corsivo
740- 830	Flash scritta
840- 860	CALL KEY d'attesa
870-1020	Assegnazione colori e caratteri per l'istogramma
1030-1070	Lettura delle DATA contenenti i nomi delle spese
1080-1160	INPUT di spese totali e assegnazione variabili principali
1170-1240	INPUT delle altre nove voci
1250-1310	Calcolo e display delle percentuali delle singole INPUT
1320-1400	Stampa e identificazione delle voci spese con loro percentuali
1410-1420	CALL KEY d'attesa
1430-1570	Stampa delle prime tre lettere del nome della spesa e istogramma
1580-1620	Stampa numerazione da zero a nove sotto i blocchi
1630-1650	CALL KEY di fine programma

VARIABILI

I	=	Variabile di ciclo FOR-NEXT principale
Z	=	Numero del conto letto nelle DATA
F	=	Linea di stampa nella simulazione della PRINT USING
M	=	Valore massimo assumibile nell'istogramma
J	=	Variabile di ciclo per colonna NUMERAZIONE ISTOGRAMMA
RI	=	Riga per la rifinitura conto
CO	=	Colonna per la rifinitura
FL	=	Variabile di controllo
IN	=	Valore numerico per la classificazione delle spese
CH	=	Valore dato a un codice carattere
BB	=	Numeratore dell'istogramma
JJ	=	Variabile di ciclo lettura cifre
TT	=	Variabile di ciclo per assunzione e assegnazione codici ASCII
N$	=	Variabile contenente il nome della voce per le INPUT
R$	=	Totale delle spese
COL	=	Variabile di ciclo per assegnazione dei colori
IST	=	Valore di base dell'istogramma
SPT	=	Spese totali
TOT	=	Valore numerico di R$
CH1	=	Codice carattere per cambiamento colore
S (I)	=	Valore numerico delle varie INPUT
P (I)	=	Percentuale delle varie INPUT
TIME	=	Variabile di ciclo temporizzatore
S (10)	=	Totale delle spese

LETTERA

Carissimi amici di TI 99 Newsoft, non immaginate nemmeno la mia felicità nel vedere finalmente in edicola una rivista interamente per il mio computer.
L'ho trovata talmente interessante che già non vedo l'ora che esca il prossimo numero.
Tralascio i complimenti per articoli (altrimenti non finirei più di scrivere) e programmi (W la CASSETTA), vi devo dire che non sono riuscito a capire l'ultima parte dell'articolo CARATTERI IN CORSIVO...

ALBERTO LINGONI

Carissimo Alberto grazie per averci fatto notare questa nostra piccola imprecisione, ma niente paura, provvederemo subito a chiarire le idee in proposito.
Scrivendo l'esempio riportato a pag. 9 del numero di giugno, otteniamo la ridefinizione della sola lettera - a - corrispondente al carattere ASCII ×97=, per ottenere tutto il set dei caratteri in corsivo, è necessario ripetere l'operazione ben 26 volte, naturalmente aggiornando ogni volta il valore del codice ASCII (97, 98, 99... 122).
Onde evitare strane maledizioni (nei nostri confronti), vi ricordiamo che se durante l'estenuante operazione di battitura commette anche un semplice SINTAX ERROOR, sarete costretti a ricominciare daccapo.
Fatte queste delucidazioni, vi auguriamo buon lavoro.

(T.S.)

CAMPAGNA ABBONAMENTI

ABBONARSI CONVIENE!

Ancora e soltanto per il mese di settembre l'abbonamento a 11 numeri di TI 99 NEWSOFT costa solo L. 80.000. In più riceverete uno splendido raccoglitore per tutta l'annata della rivista, cassette comprese!!

Affrettatevi, dunque! Non mancate questa grossa occasione.

ABBONARSI E' FACILE!

Basta inviare un assegno o vaglia postale intestato a:

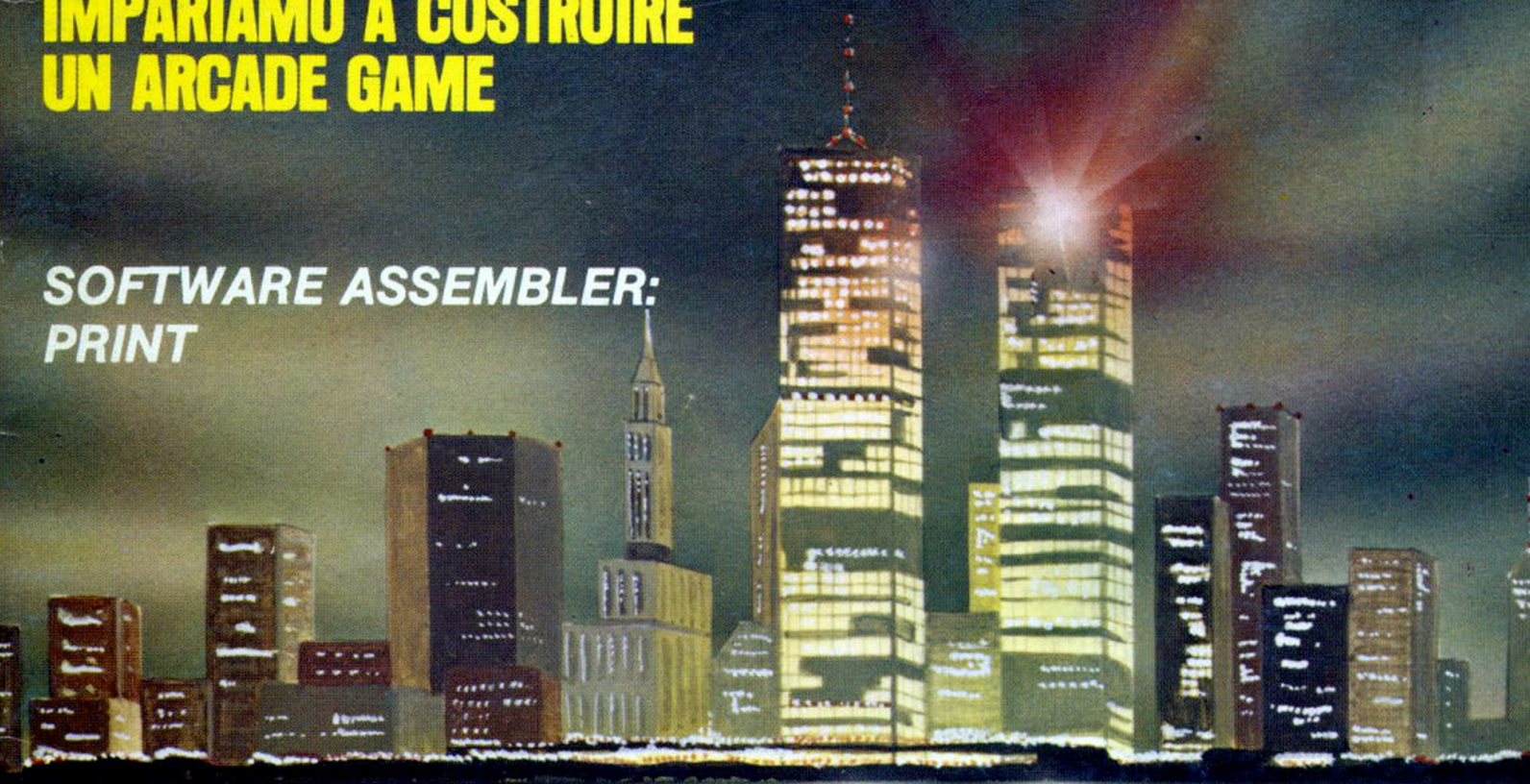
"NEWSOFT snc"
Via Stefano Jacini, 4
20121 MILANO.

DIMENTICAVAMO! *Gli abbonati potranno ottenere sconti e agevolazioni più favorevoli su alcune iniziative che presto vi "sorprenderanno".*

TI 99 NEWSOFT

OTTOBRE 1984 L. 10.000

IMPARIAMO A COSTRUIRE UN ARCADE GAME



SOFTWARE ASSEMBLER: PRINT

BASIC & EXTENDED: ISTRUZIONI DI I/O E SPRITES

10 SUPERPROGRAMMI:

Bomb on New York - The Wall - Assedio - CHAR DEF - Equo canone...

SOMMARIO

□ **UN PASSO ALLA VOLTA** pag. 2

□ **COMINCIAMO DALL'INIZIO**
Tutto quello che bisogna sapere sulle istruzioni di I/O pag. 3-4

□ **IMPARIAMO A COSTRUIRE UN ARCADE GAME: Sfondi e definizione dei caratteri**
Siamo alla seconda puntata e già in molti ci scrivono per comunicarci i "successi" raggiunti pag. 5-7

□ **DEFINIAMO I CARATTERI: CHAR DEF**
E' un programma in Extended con il quale potrete definire dei caratteri grafici pag. 7

□ **TI TEXT FILE: SFONDI CS1**
Anche questo, come il programma precedente, integra l'articolo di pag. 5. Gira in BASIC pag. 8

□ **PRIMI PASSI CON L'ASSEMBLER DEL TI 99/4A**
Continua con il modo GRAPHICS 2 la serie di articoli sull'ASSEMBLER del TI99 pag. 9-10

□ **EXTENDED: SPRITES**
Ancora SPRITES, una delle caratteristiche più interessanti del modulo SSS Extended Basic pag. 10-11

□ **BOMB ON NEW YORK**
Attenti ad essere precisi. Dovete distruggere i grattacieli di New York pag. 12-13

□ **OPERAZIONI DI SORT**
Statistica. Una delle migliori applicazioni per il nostro TI99. Nella cassetta 5 piccoli programmi per applicare quanto appreso pag. 14-15

□ **SOFTWARE ASSEMBLER: PRINT**
Pubblichiamo una, anzi due, programmi semplici, ma utili certamente ai neofiti pag. 16-17

□ **YATZEE**
Il più classico dei giochi di dadi, in un'appassionante versione per il TI99 pag. 18-19

□ **EQUO CANONE**
Possedete una casa? Volete sapere quanto pagare di affitto equo? Fate girare questo programma aggiornato fino all'anno '83 pag. 20

□ **ASSEDIO**
Un gioco ambientato nell'antico Medio Evo, trascritto per il nostro 99 pag. 21

□ **BIORITMI**
Gira in BASIC e anche in EXTENDED. Potrete calcolare rapidamente i Bioritmi per voi e i vostri amici. Occhio ai giorni critici! pag. 22

□ **LA POSTA** pag. 24

□ **PICCOLI ANNUNCI** pag. 24

TI 99 NEWSOFT

ANNO 1 - Numero due
OTTOBRE 1984
MENSILE

Direttore responsabile:
ANTONIO C. LOSITO

Coordinatore Editoriale:
VINICIO CIVININI

Art Director:
BEPPE RE-FRASCHINI

Illustrazioni:
MARIA TERESA COCOZZA

Segreteria:
EMANUELA CHIAPELLO

Redazione:
FIORENZA AURIEMMA
DARIO BAREZZI

Collaboratori:
SERGIO BORSANI,
GIANPAOLO BOTTIN,
DANIELE CATALFAMO,
PAOLO CIVARDI,
STEFANO DELLI PONTI,
FRANCESCO MACALI,
MAURO MATTIAZZI,
EZIO MONTINI,
LEVIO PEROTTI,
RICCARDO ROSSI,
MARCO SQUINTANI,
ALBERTO STRAFILE,
SEBASTIANO TOMASELLO,
PAOLO VENTAFRIDDA

TI 99 NEWSOFT è una pubblicazione della **NEWSOFT snc**, via Stefano Jacini 4 - 20121 MI
Tel. 02/807464 - Telex 324284
REGISTRAZIONE AL TRIBUNALE DI MILANO N. 323 DEL 7.7.84

© COPYRIGHT NEWSOFT snc
TUTTI I DIRITTI RISERVATI
Manoscritti e foto originali, anche se non pubblicati, non si restituiscono ed è vietata la riproduzione anche parziale di testi, fotografie e listati.

Pubblicità: AV STUDIO
Via Giuseppe Meda, 23
Tel. 02/8357155

Abbonamenti: l'abbonamento annuo a 11 numeri di *TI 99 NEWSOFT* costa L. 90.000 per l'Italia; L. 150.000 per l'Europa (spedizione aerea); L. 200.000 per tutti gli altri paesi. L'importo, per assegno non trasferibile o per vaglia postale, va inviato a NEWSOFT snc, via S. Jacini 4 - 20121 Milano.

Stampa: CGB, Albegno di Treviolo (BG)

Concessionaria per la distribuzione: EUROSTAMPA Srl - C.so Vittorio Emanuele 111 - 10123 Torino - Tel. 011/538166-7

L'editore non si assume responsabilità alcuna sul contenuto degli articoli e/o software inviati. Per gli articoli firmati e/o siglati da collaboratori esterni, la redazione si assume la responsabilità prevista dalle leggi sulla stampa.

Tutti possono collaborare a TI 99 NEWSOFT, sia con articoli che con programmi. Scriveteci o telefonateci in redazione per avere tutti i ragguagli.

UN PASSO ALLA VOLTA

Non è per ripetere sempre le stesse cose, ma – credeteci! – non ci aspettavamo di incontrare un pubblico di lettori novantanoviani così attivo *e* presente.
Fra le centinaia di lettere che ci avete mandato, molte domande richiedono una risposta sola: un passo alla volta... percarità!
C'è chi manda consigli per migliorare la rivista, aggiungendo che già così è perfetta.
Chi, invece, vuole sapere se si può realizzare una certa periferica per il TI.
E chi, non contento, vuole tutto e subito. Allora, cari amici, vi diciamo: un passo alla volta, percarità!
Le vostre richieste sono tante, mentre noi siamo pochi. Una cosa è certa, però. Attraverso le pagine della rivista risponderemo a tutti e cercheremo di fare anche l'impossibile, per quelli che ci chiedono i "miracoli".
Intanto vi lasciamo alla rivista e alla cassetta, rimandandovi al prossimo numero di novembre, che vi anticipiamo sarà "speciale"!
Dimenticavamo! Se qualcuno di voi ha un'idea o un programmino in tema "Natalizio", si faccia vivo.
NATALE deve essere una gioia per tutti, o no?

COMINCIAMO DALL'INIZIO:
Istruzioni di Input/Output

Le istruzioni basic INPUT, READ, RESTORE, PRINT, DISPLAY, costituiscono un gruppo denominato istruzioni di I/O (Ingresso/Uscita).
Esse vengono così chiamate perché la loro funzione è di ricevere e trasmettere dati da/a unità periferiche.
Il computer considera come unità periferica ogni cosa che gli permette di comunicare con il mondo esterno, un esempio può essere dato dalla tastiera o dal monitor.
Cominciamo a parlare delle istruzioni per l'inserimento dei dati, nelle quali possiamo fare un ulteriore divisione tra inserimento da tastiera ed inserimento da programma.

ISTRUZIONI DI INSERIMENTI DATI

INPUT "messaggio": variabile, variabile,...

Essa permette, come già detto, l'inserimento di dati dalla tastiera, depositando il dato inserito direttamente nella variabile specificata nel programma.
Quando il programma incontra una INPUT, ferma la sua esecuzione per riprenderla soltanto quando tutti i dati che la INPUT richiede saranno inseriti.
Il computer, normalmente, visualizza un punto interrogativo in basso a destra, ma è possibile visualizzare un messaggio (es. GIOCHI ANCORA (S-N) prima che il calcolatore scriva automaticamente il suo segnale di pronto «?», senza andare a capo, ponendolo fra virgolette prima delle variabili e facendolo seguire dai due punti (es. INPUT «GIOCHI ANCORA (S-N)»: A $ verrà visualizzato: GIOCHI ANCORA (S-N)?).
Una volta scritto il dato dovremo premere il tasto ENTER.
Quando il programma incontra una INPUT, ferma la sua esecuzione sino a quando non saranno inseriti i dati richiesti.
Se il tipo di dato inserito non è concorde con la variabile specificata, (es. introduciamo una lettera, quando è richiesto un numero), il calcolatore provvederà a segnalarci tale errore, riproponendoci il punto interrogativo.

DATA costante, costante...

In alcuni casi si rende necessario immagazzinare dati all'interno del programma, siano essi numerici o alfanumerici, per utilizzarli in seguito quando saranno necessari.
Questo immagazzinamento avviene in fase di programmazione, facendo seguire alla istruzione DATA ciò che vogliamo rimanga memorizzato.
I dati dovranno essere separati da una virgola, inoltre quelli di genere alfanumerico (es. lettere o stringhe), dovranno essere preceduti e seguiti dalle virgolette.
Tutte le costanti «dati» inserite in linee di DATA in tutto il programma, verranno considerate come se fossero su una stessa linea.
Queste costanti potranno essere lette tramite l'istruzione READ che vedremo in seguito.

ISTRUZIONI DI LETTURA DATI

PRINT

Questa istruzione visualizza il contenuto di una o più variabili, numeriche o alfanumeriche, oppure il risultato di espressioni numeriche, siano esse eseguite in modo comando o in modo programma.
L'utilizzo della PRINT in modo diretto è molto comodo perché ci permette di utilizzare il nostro computer come una semplice calcolatrice, basta far precedere la PRINT all'operazione che vogliamo eseguire; una volta premuto il tasto di conferma «ENTER», vedremo apparire il risultato direttamente sullo schermo.
In genere il computer visualizza i dati seguendo uno schema ben definito, tuttavia è possibile modificare tale uscita.
In BASIC i segni «, ; :» hanno un loro significato preciso.
Se intervalliamo le variabili da visualizzare con una virgola, i dati contenuti nelle variabili, verranno visualizzati ponendo tre spazi tra un dato e il successivo; se i dati da visualizzare non stanno tutti su

una stessa riga, verranno posti nella riga seguente, il computer continuerà in questo modo, sempre rispettando la spaziatura fino ad esaurire i dati forniti.
Il punto e virgola svolge una funzione contraria a quella della virgola, scrivendo i valori senza interporre spazi.
Ogni volta che l'interprete BASIC incontra un «due punti», dopo una PRINT, esegue una andata a capo, scrivendo il valore successivo alla linea successiva.

DISPLAY

Questa istruzione in TI BASIC esegue la stessa funzione della PRINT e segue le stesse regole.

READ variabile, variabile...

Essa permette di leggere i valori contenuti nei DATA e li trasferisce direttamente nelle variabili specificate. I dati vengono letti sequenzialmente partendo dalla prima linea di DATA. Le variabili contenute nelle READ devono essere dello stesso tipo dei dati (dati numerici con variabili numeriche e stringhe con variabili di stringa), in caso contrario verrà segnalato dal computer un errore e l'esecuzione del programma verrà interrotta.
Il numero di dati richiamati tramite la READ non deve mai superare il reale numero di dati presenti nelle linee di DATA, in caso contrario il programma si fermerà segnalando un errore.
Dopo che una READ è stata eseguita, il puntatore di DATA, indica l'ultimo dato letto, successive READ leggeranno i dati successivi a quello del puntatore.

RESTORE numero di linea

RESTORE riporta il puntatore di DATA alla prima linea di DATA. Se è specificato un numero di linea, il puntatore viene messo a partire dal primo dato presente nella linea di DATA del numero di linea specificato.

EZIO MONTINI

TI 99 MARKET è lieto di presentare una espansione di memoria da 32 K che non richiede il sistema di espansione Periferal Box.

Si collega semplicemente al computer ed è accessibile da EXTENDED BASIC, MINIMEMORY, LOGO, eccetera.

IL PREZZO E' DI 269.000 LIRE IVA COMPRESA.
PER GLI ABBONATI C'E' UNO SCONTO DEL 10%.

PER ORDINARLA IN CONTRASSEGNO:

TI 99 MARKET
Via Stefano Jacini, 4 - 20121 MILANO

IMPARIAMO A COSTRUIRE UN ARCADE GAME:

Sfondi e definizione dei caratteri

Normalmente per costruire dei Video Giochi si parte sempre dal disegno dello sfondo. Questo è naturalmente valido per i giochi da bar, mentre per i giochi in casa, normalmente si tende ad evitare questa parte per due motivi: la memoria occupata e la perdita di tempo. Nell'elenco dei motivi, non abbiamo volutamente incluso il fatto della ovvia difficoltà nella creazione di sfondi in alta risoluzione. Per potere creare degli sfondi, è necessario stabilire a priori, il tipo di sfondo che si vuole ottenere. In base alla storia che con il gioco si vuole raccontare, lo sfondo potrà essere fisso, o mobile.

In questo articolo parleremo degli **sfondi fissi** mentre lasceremo gli sfondi mobili ad un prossimo articolo.

Dopo avere stabilito che lo sfondo è fisso, bisogna disegnare e colorare su carta lo sfondo, prima di convertire in codici esadecimali i caratteri che lo compongono.

A questo proposito può tornare utilissimo un foglio di carta millimetrata suddivisa in quadretti di 8x8 pixel. L'optimum sarebbe se il foglio contenesse tutto lo schermo, così da potere comporre lo sfondo completo. Sul foglio, una volta muniti di gomma e matita, ci si può scatenare con le proprie velleità artistiche.

Purtroppo per voi dobbiamo spegnere sul nascere l'entusiasmo poichè per i caratteri siamo vincolati da quelli a disposizione. Cominciamo a togliere i caratteri che rappresentano i numeri, poichè servono per i punti, poi bisogna togliere i caratteri che useremo per comporre i nemici, i proiettili, e il nostro «Eroe». Alla fine ci accorgiamo di avere un centinaio di caratteri malcontati contro i 768 che sono da riempire. A questo punto, l'unica è comporre degli scenari componendoli con dei mattoni costituiti da caratteri predeterminati, per esempio comporre un carattere pieno uno vuoto e tre a 45°, mentre gli altri verrebbero usati per caratteri particolari, sempre usando il tutto con parsimonia.

Un esempio di questa tecnica, è stato usato per comporre lo sfondo della routine dei nemici (nel numero scorso), quindi basterà dare una scorsa al listato per potere verificare l'efficacia del sistema, nonchè la semplicità del tutto.

Una volta composta *l'opera d'arte*, bisogna tirare fuori dai disegni i codici esadecimali che li compongono. Ben conoscendo l'abilità dei nostri lettori come emanuensi, siamo sicuri, che se l'impresa a parole è molto facile, anche per loro la realtà è ben diversa. Quindi alleghiamo all'articolo di questo mese, non una Routine ma bensì una **Utility per comporre i caratteri**. La troverete più avanti nella rivista, in cassetta e listato. Basti comunque sapere che questo programma vi faciliterà enormemente, poichè oltre a dare il codice esadecimale del carattere, dà la possibilità di vederlo in inverso e di modificarlo, di comporlo in call magnify (3) (insieme ad altri 3 caratteri) a vederlo insieme ad altri uguali, cancellarli, usare i cursori in diversi modi ecc. Insomma, un programma tutto da scoprire, da notare che gira anche in *Extended.*

Ed ora il punto dolente: la stampa del risultato.

Un primo avviso che vogliamo darvi, è dato dal fatto che in questa parte del programma si annideranno tanti bug (errori) per cui sarebbe buona norma comporre prima lo sfondo e farlo girare, poi «attaccarci» il resto del programma.

Poichè la composizione di una pagina grafica porta via molta memoria, anche qui bisogna ingaggiare contro il computer una battaglia all'ultimo byte. In ogni caso anche qui i soliti consigli primo fra tutti, il più importante, usare senza parsimonia il DATA-READ-RESTORE. Infatti, è in queste procedure grafiche che se ne apprezza maggiormente l'utilità: due sono i modi utilizzo.

Primo modo di utilizzo, nel

caso in cui il campo da coprire copre più del 50% dello schermo, consiste nel trasformare lo sfondo in stringhe, inserirle nei data e poi con un ciclo FOR-NEXT, altro grande protagonista della grafica, caricarle e stamparle direttamente:

```
100 CALL CLEAR
110 CALL CHAR (40, «0»)
120 CALL CHAR
    (41, «FFFFFFFFFFFFFFFF»)
130 CALL CHAR
    (48, «0103070F1F3F7FFF»)
140 CALL CHAR
    (49, «80C0E0F0F8FCFEFF»)
150 CALL CHAR
    (50, «FF7F3F1F0F070301»)
160 CALL CHAR
    (51, «FFFEFCF8F0E0C080»)
170 DATA 0)))))1,0)))))))
    1,0)))))))))1,0)))3   2)))1
180 DATA )))3   2))),))) 0)1
    ))),))) ))) ))),))) 2)3   )))
190 DATA )))1 0))),2)))1
    0)))3,2)))))))))3
200 DATA 2)))))))3,2)))))3
205 R=11
210 FOR C=6 T010
220 R=R—1
230 READ A$
240 FOR K=1 TO LEN(A$)
245 GOSUB 450
250 CALL HCHAR (C, R+K, KK)
270 NEXT K
280 NEXT C
290 FOR C=11 TO 17
300 READ A$
310 FOR K=R+1 TO LEN(A$)
320 GOSUB 450
330 CALL HCHAR(C, K, KK)
340 NEXT K
350 NEXT C
355 R=6
360 FOR C=18 T0 21
370 R=R+1
380 READA$
390 FOR K=1 T0 LEN(A$)
400 GOSUB 450
410 CALL HCHAR(C, K+R, KK)
420 NEXT K
430 NEXT C
440 GOTO 440
450 KK=ASC(SEG$(A$, K, 1))
460 RETURN
```

Altrimenti, nel caso si tratti di figure piccole si può inserire nei DATA i parametri di stampa da inserire nei CALL HCHAR:

```
100 CALL CLEAR
110 DATA 2,3,4,5,6,7,8,9,10,11,12
120 A$=«TI99 NEWSOFT»
130 FOR C=1 T0 LEN(A$)
140 READ F
150 KK=ASC(SEG$(A$, C, 1))
160 CALL HCHAR(12, F, KK)
170 NEXT C
180 END
```

La stringa «A$» in questo caso usata per l'esempio verrà chiaramente sostituita da una stringa grafica.
Inoltre, sempre per risparmiare memoria, rimanendo nel contesto precedente, ricordate di usare per i cicli FOR-NEXT sempre le stesse variabili, di definire variabili con il nome più corto possibile, usare il minor numero di REM, o non utilizzarle addirittura. Insomma dovete ridurre lo spreco al massimo poichè bisogna tener presente che la grafica consuma, anzi mangia, la memoria velocemente.
A questo proposito, sarebbe possibile, nel caso vi fosse la necessità, trasformare le stringhe dello sfondo, in FILE e memorizzarle su cassetta tramite un programma a parte, e poi farle richiamare dal programma che le utilizza facendole solamente transitare nella memoria del vostro TI 99 giusto il tempo necessario a stamparle.
In questo modo si possono fare anche dei giochi multischermo, nella cassetta allegata alla rivista troverete un breve programma per il caricamento delle stringhe in CS1. (Andate a leggere anche l'articolo a pag. 8).
Con questo programma è possibile memorizzare su cassetta i vostri sfondi o comunque le vostre figure. Per richiamarle, basta inserire queste brevi righe nel listato del programma che deve uti-

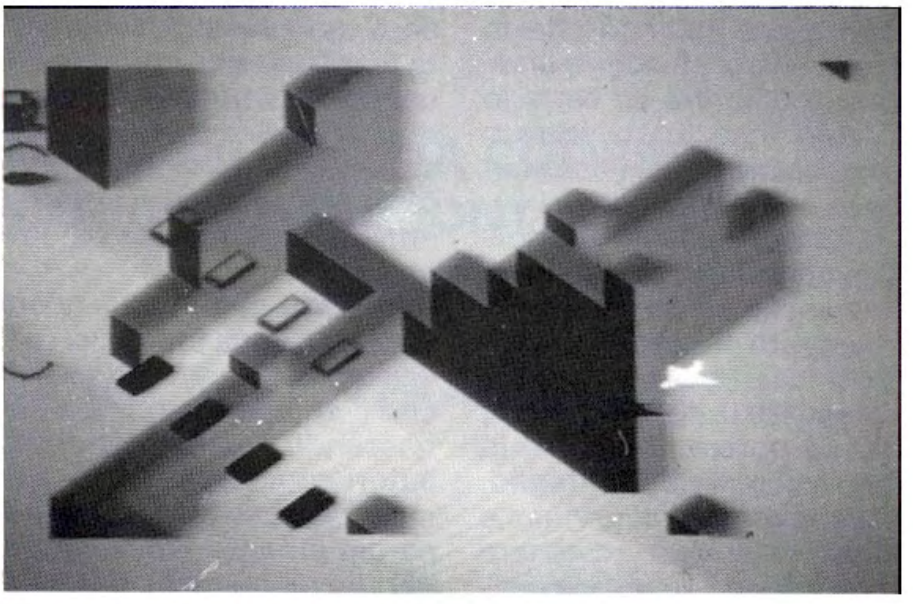


Immagine tratta dal gioco «ICE-WALL BATTLE 3-D.»: lo sfondo a 3 dimensioni scorrevole, tipo ZAXXON, è stato costruito come descritto nell'articolo.

lizzare la grafica archiviata:

```
10 OPEN#1:«CS1», SEQUENTIAL,
   INPUT, INTERNAL, FIXED 128
20 CALL CLEAR
30 .....
1000 INPUT#1:A,B
1010 FOR C=B TO A+B—1
1020 INPUT#1:A$
1030 CALL CHAR(C,A$)
1040 NEXT C
1050 INPUT#1:F
1060 FOR C=1 TO F
1070 INPUT#1:A$
1080 FOR D=1 TO 32
1090 CALL HCHAR(C, D, SEG$
     (A$, D, 1))
1100 NEXT D
1110 NEXT C
1120 RETURN
```

E questo era per i FILE, e per chi ama la vita spericolata.
Un'ultimo consiglio. Sovente, ci è capitato di vedere, gli sfondi, memorizzati in matrici!!!
Il nostro consiglio e di non usare questo sistema se non è strettamente necessario, poichè anche se è più veloce, occupa il doppio della memoria occorrente a memorizzare stringhe con i sistemi che nell'articolo sono trattati.
Dopo questa nostra lunga chiacchierata non pretendiamo che siate diventati degli esperti GAME-DESIGNER. Bensì speriamo che queste nostre esperienze servano a qualcuno di voi affinchè possiate per lo meno adoperare meglio la vostra macchina.
Con questo non resta che darci appuntamento sempre su queste pagine il mese prossimo per parlarvi degli **sfondi mobili o scorrevoli** che dir si voglia, e per adesso vi lasciamo tutto il tempo per impegnarvi con i nostri programmi.

MAURO MATTIAZZI
(MMG)

DEFINIAMO I CARATTERI: CHAR DEF

Il programma che alleghiamo questo mese all'articolo, contrariamente a quanto scritto il mese scorso, è un programma di utilità.

Infatti questo programma vi potrà tornare utile ogni qualvolta dovrete definire dei caratteri grafici o comunque per ricavare i codici esadecimali degli SPRITE.

Il programma è indubbiamente molto complesso ma ogni caso con queste istruzioni e con le legende inserite nel programma il suo uso è notevolmente semplificato.

Per prima cosa si devono decidere i colori uno per il background (sfondo) e uno per (primo piano) foreground.
A questo punto sul video apparirà un'area quadrata grande che chiameremo **area di lavoro** e quattro aree molto più piccole che serviranno per **visualizzare i caratteri singoli** uno per area ed infine quattro caratteri per **visualizzare lo SPRITE completo**.
Sotto l'area grafica viene stampata la tabella dei comandi. Diamo qui di seguito una sommaria spiegazione dei comandi, rimandando il nostro lettore all'uso del programma per meglio comprendere la comodità e la versatilità di tale programma.

0: con questo comando si fà avanzare il cursore all'interno dell'area di lavoro lasciando i pixel spenti.
1: funziona come lo «0» con la differenza che i pixel li accende.
E: serve per spostare velocemente il cursore nel senso della freccia.
S: come per la «E»
X: come per la «E»
D: come per la «E»
K: serve quando il carattere composto nell'area di lavoro è completo, in modo da permettere al calcolatore di determinare il codice esadecimale e di stamparlo in una delle quattro aree da noi scelta.
I: serve per ottenere il carattere in quel momento raffigurato nell'area di lavoro in inverso, cioè a colori invertiti.
H: serve per portare il cursore in posizione 1,1.
C: serve per pulire l'area di lavoro.
R: serve per richiamare uno dei caratteri in quel momento memorizzati nelle quattro aree appositamente predisposte;
P: opzione libera ossia premendo «P» il programma va ad una linea vuota che può essere definita dall'utilizzatore. Un'idea potrebbe essere quella di inserire la memorizzazione dei codici esadecimali sul «CS1» in modo da creare un aggancio con l'utility che accompagna l'articolo, cioè quello che vedremo nel prossimo numero parlando dei fondi scorrevoli.
M: con questo comando potete decidere se il cursore dopo che avete premuto «1» o «0» dovrà scorrere di una posizione avanti in orizzontale oppure no.
Z: poichè il computer dopo che avete scelto il colore di foreground e di backround, sceglie tra i restanti colori un colore scuro per lo sfondo, ed un colore chiaro per le scritte, può darsi che le scelte del TI non siano di vostro gradimento, così, premendo «Z» si cambiano i colori. Lo si può usare in qualunque momento e tutte le volte che volete.
T: serve per selezionare il carattere nella rappresentazione ridotta o meglio in modo alfabetico.
N: serve per ritornare ad avere il carattere normale. È quindi l'esatto opposto del comando «T».

(MMG)

TI TEXT-FILE: SFONDI CS1

Il programma «SFONDI CS1» permette di inserire i codici ESADECIMALI voluti, ed in seguito, di comporre lo sfondo con i caratteri ridefiniti.
Sulla memoria di massa, il **registratore**, verranno memorizzate sia le stringhe che compongono lo schermo (lunghe 32 caratteri) sia i codici esadecimali, nonchè tutti i dati necessari al TI affinchè sia in grado di ricostruire le immagini in precedenza memorizzate.
Il programma, gira in TI BASIC, per l'EXTENDED invece bisogna apportare le seguenti modifiche:

```
120 FOR C=120 TO 136
150 FOR V=1 TO 14
200 IF A>104 THEN 190
210 B=87-(A-49).
```

Il programma è stato strutturato in modo che i caratteri che rappresentano le lettere ed i numeri vengano usati per ultimi.
I comandi sono pochi e semplicissimi.
Al RUN ci sarà la definizione dei colori e l'apertura dei FILE (righe 100a 180), dopo di che il computer vi chiederà quanti sono i caratteri da definire, e predisporrà un ciclo FOR-NEXT opportunamente dimensionato (190 a 240).
Ogni codice verrà preceduto da un quadretto vuoto, che dopo l'«ENTER» del codice prenderà la forma definita dalla stringa esadecimale appena digitata (250 a 270). Se la figura non è quella da voi voluta, è possibile ridefinire il carattere, rispondendo «N» alla domanda della macchina di memorizzare sul supporto magnetico. Altrimenti, basterà premere «Y» e la stringa verrà inserita nell'archivio (280 a 330).
Definiti tutti i caratteri, si passa alla composizione dello sfondo.
Poichè il computer non parte mai dallo stesso carattere per definire la grafica, vi daremo il sistema per sapere quali tasti premere, in ogni caso, è bene, man mano che si procede, annotare le lettere corrispondenti alle figure, in modo da sveltire il lavoro.
Per sapere dove partire basta ricorrere alla seguente tabella:

Qta CHR Def.	2	3	4	5	6	7	8	9	10	11	12	13	14
CTRL +	U	T	S	R	Q	P	O	N	M	L	K	J	I

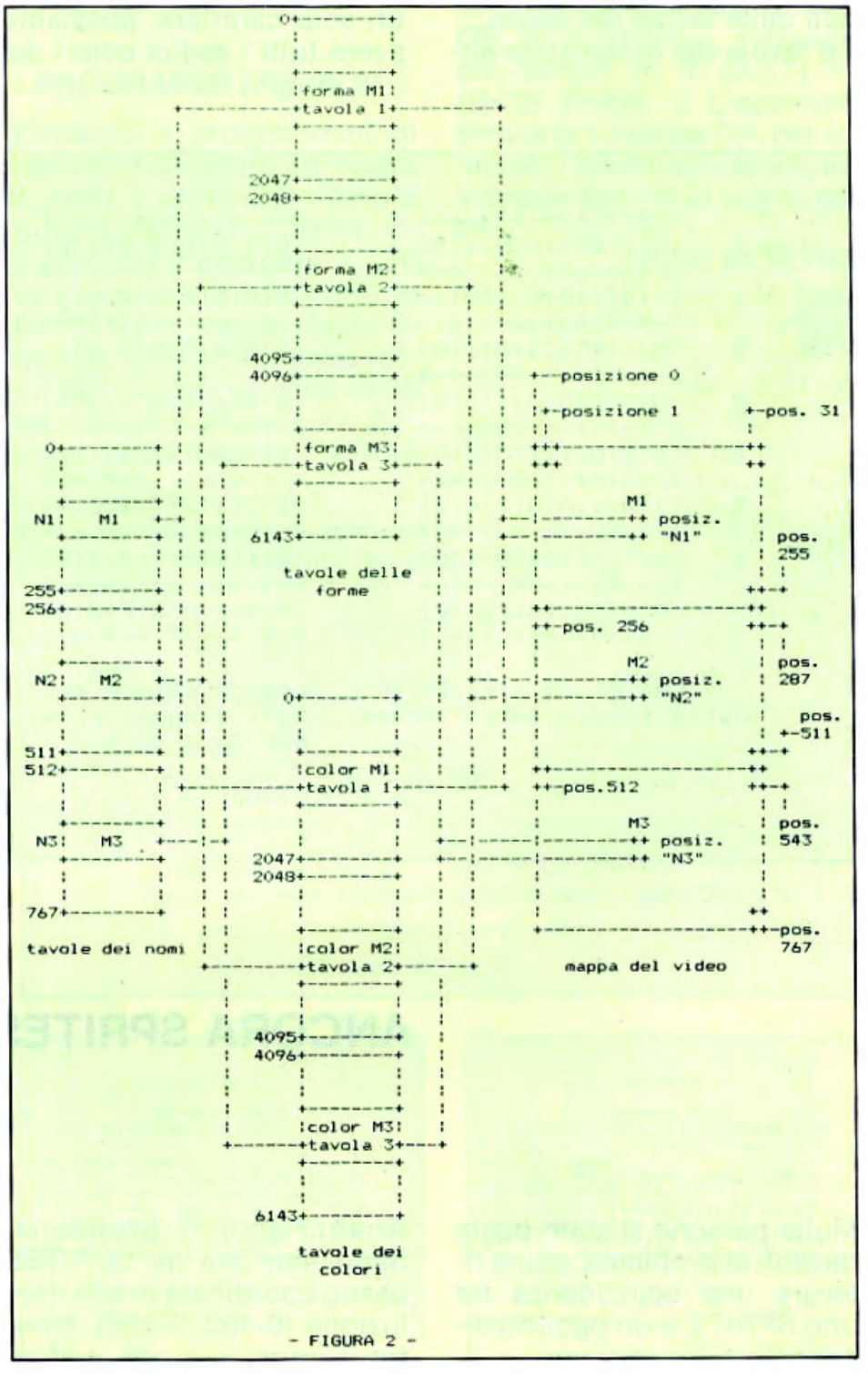
Sapendo quanti caratteri avete definito, basta guardare la lettera corrispondente e saprete da che carattere partire per la stampa, naturalmente premendo insieme alla lettera il tasto CTRL.
Nel caso il tasto non sia stato definito, un segnale acustico vi segnalerà l'errore di battitura; mentre se erroneamente inserirete un carattere errato, basterà battere «SHIFT-1» e subito tornerete indietro di una posizione.
In questa seconda parte del programma, il computer vi chiederà di quante stringhe sarà composta la figura che volete comporre. Anche qui verrà dimensionato un ciclo FOR-NEXT e, memorizzato tale numero (360 a 390), tramite gli appositi tasti comporrete le stringhe e le farete memorizzare.
Finito il lavoro, potrete replicare tutte le volte che vorrete.

(MMG)

PRIMI PASSI CON L'ASSEMBLER DEL TI-99/4A

GRAPHICS 2

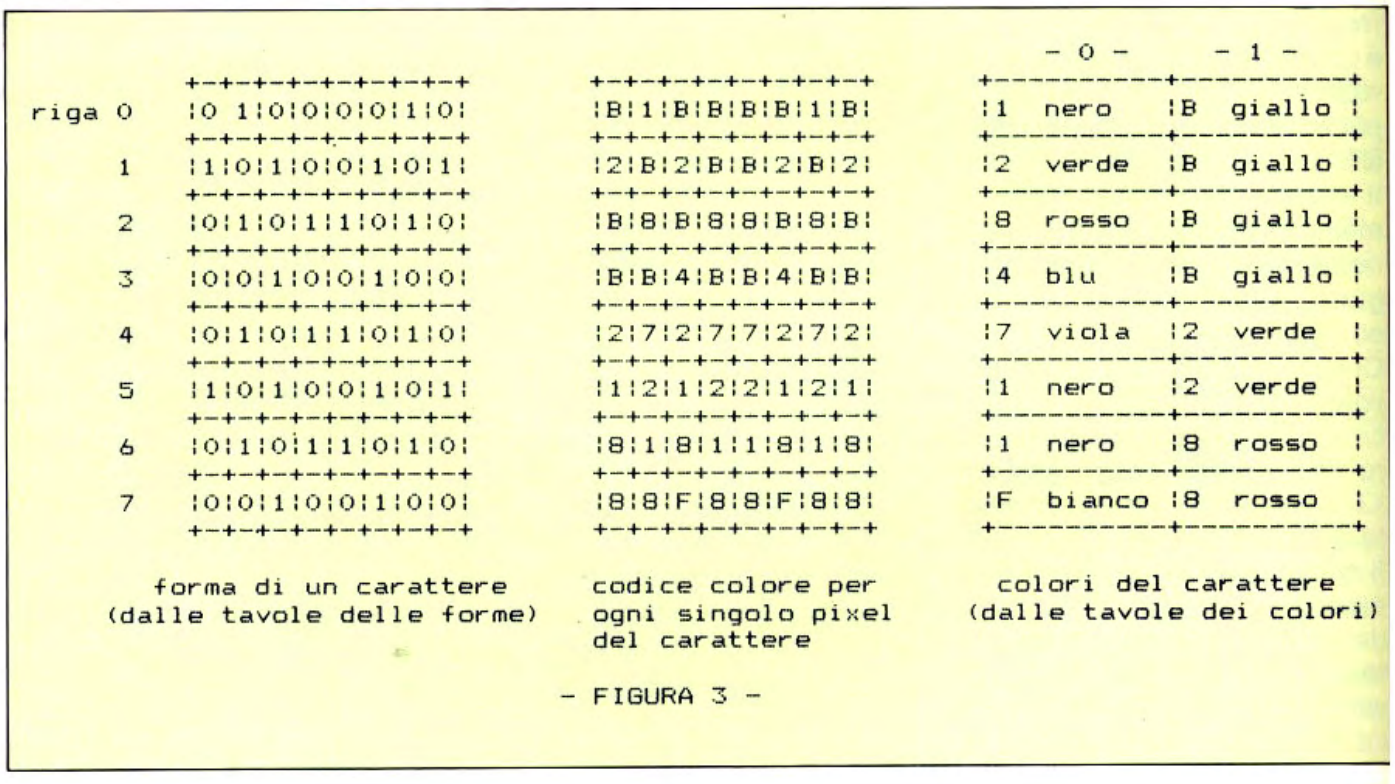
Nel primo numero, abbiamo descritto <GRAPHICS 1>, che è il modo utilizzato più frequentemente, passiamo ora alla descrizione di <GRAPHICS 2>, o più semplicemente <BIT MAP MODE>.
Il modo grafico 2 è molto simile al modo grafico 1, dispone però di una quantità maggiore di memoria per le forme ed i colori dei caratteri.
Come GRAPH. 1, ha una tavola video di 768 «posizioni carattere» in cui memorizzare un numero compreso tra <0> e <255> corrispondente al nome del carattere.
Il nome, come ricorderete, rimanda ad una tabella (tavola delle forme) in cui si trova la forma del «pattern» rappresentata da 8 byte.
La differenza sostanziale con GRAPH. 1, è che esistono ben tre tavole delle forme ognuna delle quali, si riferisce ad un determinato segmento dello schermo che, per capire meglio, dobbiamo immaginare diviso in tre parti uguali in senso orizzontale. Ognuna delle tre parti, è formata da 256 «posizioni carattere», che fanno riferimento alal corrispondente tavola delle forme, la prima parte di schermo, quello in alto, fa' riferimento alla prima tavola, e di seguito, le altre.



- FIGURA 2 -

In pratica, questo significa, che ogni singola «posizione carattere» può contenere un carattere di forma diversa dagli altri, e che uno stesso nome, ha una forma diversa se si trova nel primo, nel secondo, o nel terzo segmento del video.
La memoria occupata dalle tavole delle forme è 6144 byte, tanti quanti sono utilizzati dalle tavole dei colori.
Le tavole dei colori sono anch'esse tre, ognuna abbinata alla rispettiva tavola delle forme. Vediamo ora come sono organizzate.
A differenza di GRAPH. 1, che permetteva di definire solo due colori per ogni gruppo di otto nomi, GRAPHIC 2, permette di definire ben sedici colori per ogni singolo nome, in poche parole significa che nello spazio di un solo carattere, possiamo avere tutti i sedici colori del nostro computer!.
Questi sedici colori sono suddivisi in due per ogni riga di otto pixel, nella figura 3 troverete una rappresentazione grafica di come avviene la visualizzazione di un carattere, mentre in figura 2 troverete la mappa completa di GRAPHICS 2.

E. MONTINI
S. TOMASELLO



	forma di un carattere (dalle tavole delle forme)	codice colore per ogni singolo pixel del carattere	colori del carattere (dalle tavole dei colori)
			- 0 - / - 1 -
riga 0	0 1 0 0 0 0 1 0	B 1 B B B B 1 B	1 nero / B giallo
1	1 0 1 0 0 1 0 1	2 B 2 B B 2 B 2	2 verde / B giallo
2	0 1 0 1 1 0 1 0	B 8 B 8 8 B 8 B	8 rosso / B giallo
3	0 0 1 0 0 1 0 0	B B 4 B B 4 B B	4 blu / B giallo
4	0 1 0 1 1 0 1 0	2 7 2 7 7 2 7 2	7 viola / 2 verde
5	1 0 1 0 0 1 0 1	1 2 1 2 2 1 2 1	1 nero / 2 verde
6	0 1 0 1 1 0 1 0	8 1 8 1 1 8 1 8	1 nero / 8 rosso
7	0 0 1 0 0 1 0 0	8 8 F 8 8 F 8 8	F bianco / 8 rosso

- FIGURA 3 -

ANCORA SPRITES

Molte persone si sono poste davanti al problema: come rilevare una coincidenza tra uno SPRITE e un oggetto disegnato sullo schermo.

Analizziamo il problema. Sappiamo che gli SPRITES usano coordinate in alta risoluzione (0-192, 0-256), mentre il normale modo grafico usa coordinate (1-24, 1-32). Non esiste istruzione BASIC che si occupi di rilevare tale coincidenza, però esistono le istruzioni CALL POSITION e

CALL GCHAR. La prima si occupa di rilevare la posizione dello SPRITE specificato, la seconda di rilevare se in una determinata posizione dello schermo è presente un determinato carattere.
La CALL POSITION restituisce le coordinate in modo «alta risoluzione» (0-192, 0-256), quindi prima di poter essere utilizzate per una qualsiasi operazione che non riguardi gli SPRITES, dovranno essere divise per otto, per ottenere un numero di linea compreso tra 1 e 24, e un numero di colonna tra 1 e 32.
Per poter risolvere il nostro problema dovremo utilizzare una CALL POSITION, le coordinate che ci verranno restituite, opportunamente divise ci daranno la posizione dello SPRITE, nei confronti dello schermo grafico normale.
Introducendo i risultati della nostra divisione in una CAL GCHAR, rileveremo se lo SPRITE è venuto in contatto con il carattere da noi preso in considerazione, controllando la variabile di ritorno.
Dobbiamo tenere in considerazione il fatto che le coordinate dello SPRITE sono riferite all'angolo in alto a sinistra, quindi per ottenere una sicura coincidenza è consigliabile utilizzare in fase di costruzione di programma le formule di coversione riportate nella figura 1.
Un fattore molto importante per la buona riuscita dell'operazione e rappresentato dalla velocità dello SPRITE nel momento della coincidenza.
A causa dei calcoli e delle verifiche che il computer deve effettuare, la coincidenza non può essere rilevata se la velocità è troppo elevata.
Dovremo quindi costruire e strutturare il programma a seconda della velocità che vogliamo imprimere allo SPRITE.
Velocità superiori ai 40 di riga o di colonna non daranno mai risultati ottimali, i due programmini di esempio dimostrano la strutturazione di un programma tipo per velocità da 1 a 35 nel 1 caso, da 1 a 40 nel secondo.

I PROGRAMMI (Figura 2)

Entrambi i programmini di esempio mettono in opera quanto è stato detto sino a questo momento. Verrà visualizzato il carattere ASCII 35 «÷» nel centro del video, lo SPRITE posto sulla stessa riga si sposterà da sinistra verso destra con velocità di 35 nel caso 1 e 40 nel caso 2.
Ogni volta che lo SPRITE (rappresentato dall'ASCII 42 «*») toccherà il carattere «÷», verrà emessa una nota.
Il secondo programmino e più compresso per permettere una maggiore velocità di esecuzione, necessaria per rilevare le coincidenze alla velocità di 40.
Nota: Gli operatori logici AND, OR, richiedono un maggior tempo di esecuzione se specificati, (es. IF A<1 OR B<2 THEN..., richiede più tempo di IF (A<1) + (B<2) THEN...). L'esempio simula la funzione OR, per simulare l'AND basta sostituire il segno **per** «*» al segno **più** «+».

EZIO MONTINI

TABELLA DI CONVERSIONE

DA	A	MODO
NRIGA	GRIGA	(NR*8)-2
NCOLONNA	GCOLONNA	(NC*8)-2
GRIGA	NRIGA	(GR/8)+2
GCOLONNA	NCOLONNA	(GC/8)+2

NR= RIGA NORMALE GR= RIGA GRAFICA
NC= COLONNA NORMALE GC= COLONNA GRAFICA

Figura - 1 -

Es. No 1

Extended BASIC
Velocita' massima ottenibile : 35

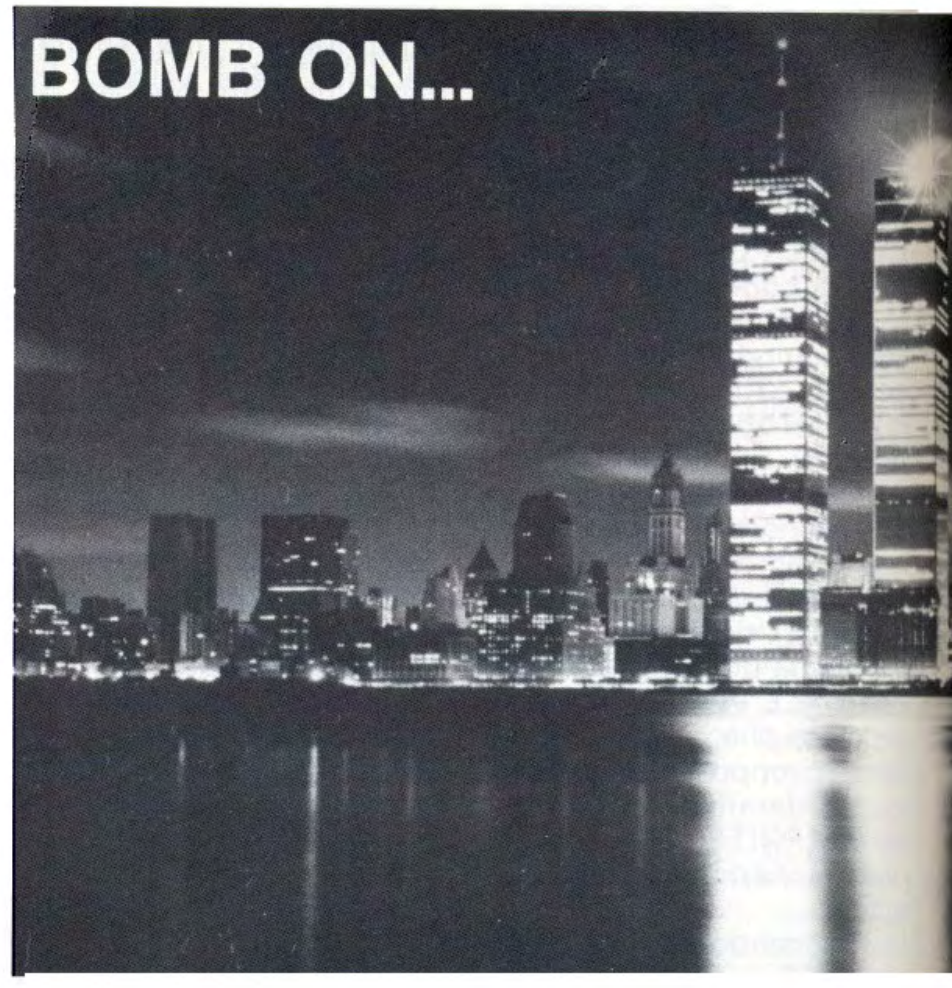
```
100 CALL CLEAR
110 CALL SPRITE(#1,42,2,94,9,0,35)
120 CALL HCHAR(12,16,35)
130 CALL POSITION(#1,X,Y)
140 IF X<8 OR Y<8 THEN X,Y=8
150 CALL GCHAR(X/8,Y/8,R)
160 IF R=35 THEN CALL SOUND(110,110,0)
ELSE 130
170 GOTO 130
```

Es. No 2

Extended BASIC
Velocita' massima ottenibile : 40

```
100 CALL CLEAR
110 CALL SPRITE(#1,42,2,94,9,0,40)
120 CALL HCHAR(12,16,35)
130 CALL POSITION(#1,X,Y) :: IF (X<8)+
(Y<8) THEN X,Y=8
140 CALL GCHAR(X/8,Y/8,R) :: IF R=35
THEN CALL SOUND(110,110,0) ELSE 130
150 GOTO 130
```

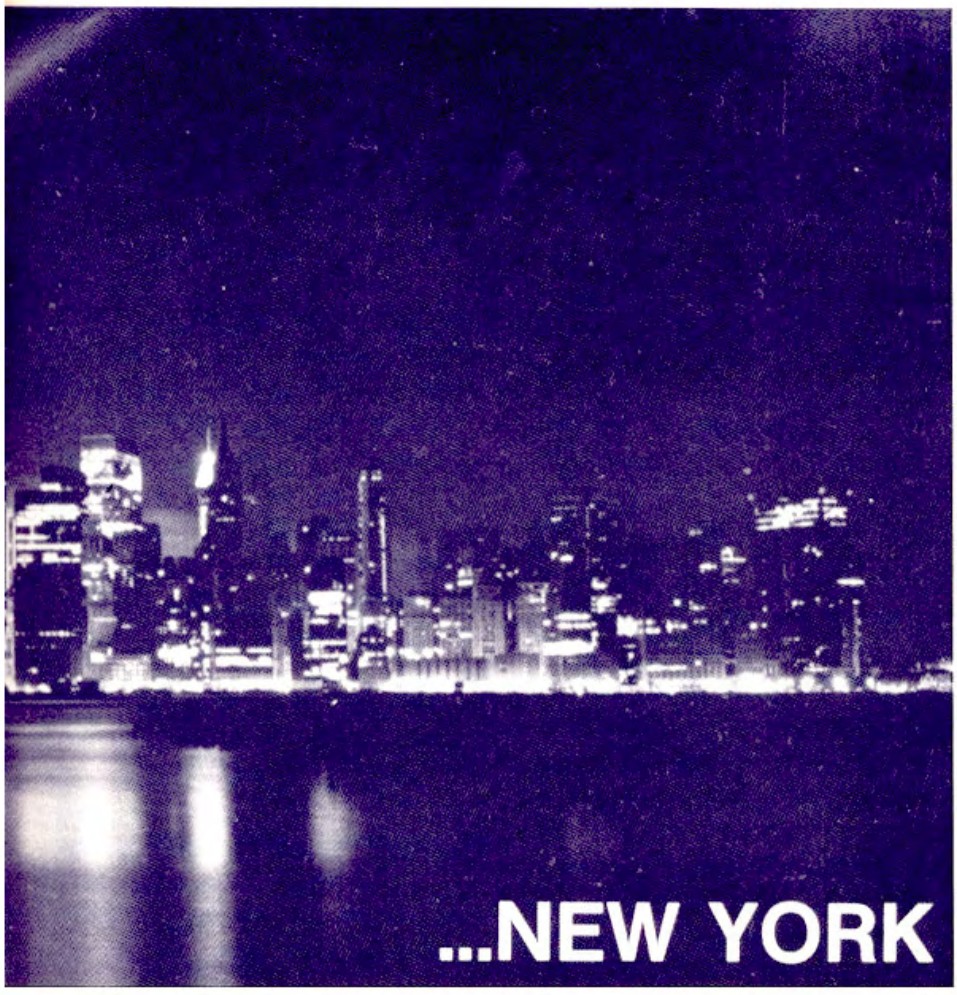
Figura - 2 -



Ma qui i russi non c'entrano; siete voi che invece dovrete far sfoggio della vostra abilità, riflessi e mira per riuscire ad atterrare nell'area originariamente occupata dai massicci e pericolosi grattacieli. Come fare?
È presto detto. Partite dall'angolo in alto a sinistra del video ed avanzate inarrestabilmente verso la sponda opposta col vostro jet. La difficoltà consiste nel fatto che una volta raggiunta l'altra estremità dello schermo, comincerete a scendere di quota. Dovrete quindi avere la preoccupazione ad ogni passaggio di abbattere quanti più piani potete dei sempre più vicini «skyscrapers», in modo d'avere sempre un certo spazio vitale che consenta l'atterraggio finale.
Il supersonico può sganciare una sola bomba per volta e non può sganciarne una seconda finchè la prima non abbia colpito l'obiettivo o toccato il suolo. La bomba si sgancia con la barra-spazio e se tocca il terreno vi fa perdere dei punti.
Il punteggio ottenuto dalle incursioni appare visualizzato nella parte alta dello schermo. L'abbattimento di un piano di grattacielo vale cinquanta punti e in caso di completamento della missione (atterraggio) avrete un bo-

REMARKS

190 - 210	Stampa opzioni orario missione
220 - 270	CALL KEY per scelta colore di fondo
280 - 380	Assegnazione colore schermo e stereo
390 - 420	Assegnazione variabili principali
430 - 620	Assegnazione colori e grafica
630 - 760	Stampa schemata principale
710 - 810	Stampa messaggi
820 - 1020	Ciclo randomizzato costruzione grattacieli
1030 - 1060	Assegnazione variabili di posizione
1070 - 1130	Stampa punti sullo schermo di gioco
1140 - 1200	Controllo posizione aereo per atterraggio e suoni
1210 - 1320	Stampa movimento aereo
1330 - 1360	CALL KEY per sganciamento bomba
1370 - 1470	Sganciamento bomba e sua caduta
1500 - 1520	Distruzione di un piano con accredito dei 50 punti
1530 - 1640	Controlli per mancata distruzione di un piano di grattacielo
1650 - 1760	Spostamento aereo durante sganciamento bomba
1770 - 1800	Collisione con grattacielo
1810 - 1890	Stampa in caso fosse rimasto un solo aereo e punteggio aggiornato
1900 - 1950	Stampa in caso fossero rimasti due aerei e punteggio
1960 - 1970	Stampa in caso fossero finiti gli aerei e richiesta nuova partita
1980 - 2010	CALL KEY per continuazione o meno della partita
2020 - 2060	Stampa saluto di fine gioco
2070 - 2090	Ciclo temporizzatore e fine gioco (END)
2100 - 2210	Subroutine «PRINT AT»
2220 - 2270	Congratulazioni per la buona riuscita dell'atterraggio
2280 - 2290	Ciclo temporizzatore
2300 - 2310	Punti extra per l'atterraggio



LISTA VARIABILI

A	=	Riga per la visualizzazione relativa al jet
B	=	Colonna per la visualizzazione del jet
K	=	Variabile di ritorno nelle CALL KEY
S	=	Variabile di Status nelle CALL KEY
Y	=	Variabile contenente il colore dello schermo
G	=	Gruppo caratteri usato nelle CALL COLOR
I	=	Variabile che dà il codice ASCII del carattere contenuto nella stringa A$
C	=	Colonna del grattacielo
K	=	Riga del grattacielo
A	=	Scelta randomizzata del colore del grattacielo
H	=	Colonna usata per la visualizzazione dei punti sullo schermo di gioco
P	=	Variabile contenente il numero dei punti totalizzati
J	=	Carattere controllato nelle CALL GCHAR (per l'aereo)
F	=	Riga per la visualizzazione della bomba
M	=	Colonna per la visualizzazione della bomba
E	=	Carattere controllato nelle CALL GCHAR (per la bomba)
T	=	Variabile di tempo
R	=	Trasferisce il carattere letto in un valore numerico utile per la stampa
C	=	Trasferisce il carattere letto nella CALL HCHAR come colonna
L	=	Variabile che contiene, dal quinto in poi, il carattere contenuto in A$
AI	=	Codice del carattere letto da L in poi
A$	=	Stringa contenente i vari tipi di messaggi
CH	=	Attribuisce il codice carattere da visualizzare nel grattacielo
H$	=	Valore in stringa di P
R$	=	Lettura del primo e secondo carattere contenuti in A$
C$	=	Lettura del terzo e quarto carattere contenuti in A$
JET	=	Numero di aerei
INF	=	Variabile di controllo per bonus aereo aggiuntivo
FLAG	=	Variabile di controllo per rinnovamento punteggio
CA	=	Variabile controllo colore aereo

nus di duemila punti. Se scendete sotto il livello del grattacielo più alto andrete inevitabilmente a sbattere perdendo uno dei tre jet a vostra disposizione. La perdita di un aereo tuttavia è relativa perché raggiunto il punteggio di diecimila o cinquantamila ve ne verrà accreditato un altro. Lo svolgersi del gioco è sottolineato da suoni generati dall'ottimo microprocessore audio di cui il TI 99 è dotato.

L'altezza dei grattacieli, la loro disposizione e il loro colore sono interamente affidati alla casualità di due istruzioni randomizzate che ne regolano la configurazione. Altra variante interessante di «Bomb on New York» è la possibilità di poter scegliere l'ora della missione: mattino, pomeriggio, sera o notte, resa possibile dalla combinazione di una CALL KEY e di una CALL SCREEN.

Per chi non riuscirà nell'intento, inutile sottolinearlo, sarà notte fonda, ovvero il beffardo «vuoi giocare ancora»?

MARCO SQUINTANI

OPERAZIONI DI SORT

Nel numero precedente, nel corso di una breve introduzione alla Statistica, avevamo accennato all'opportunità di disporre di operazioni di ordinamento di dati, operazioni che, con un termine inglese diventato di uso comune, vengono chiamate di Sort.
Il sort è utile sia nel caso in cui i dati da ordinare siano numerici che quando questi dati sono letterali.
Intuitivamente, il primo sistema che ci viene in mente per effettuare una ricerca è quello sequenziale, leggendo cioè una lista dall'inizio fino a ritrovare il nome cercato.
Questo tipo di approccio è molto semplice ma la ricerca potrebbe essere molto lunga nel caso che i nomi, o i dati, siano molti; mediamente in una ricerca sequenziale il tempo necessario per ritrovare un nome è direttamente proporzionale al numero totale di dati presenti in una lista.
Ammettendo di avere 1000 dati, sarà necessario eseguire, nel caso peggiore, 500 confronti prima di poter terminare con la conclusione che il nome da noi cercato non esiste.
Descriveremo cinque metodi di sort (in ordine ascendente e discendente); inoltre poiché ai fini di un eventuale utilizzo statistico è sovente necessario comparare coppie di valori, alcuni degli algoritmi proposti saranno in grado di ordinare i dati a coppie.

CONSIDERAZIONI GENERALI

Ci si potrebbe attendere che esista un modo ottimale per ordinare e ritrovare nel più breve tempo possibile i dati in modo da scrivere una volta per tutte questo algoritmo nei nostri programmi e risparmiare del tempo prezioso.

Purtroppo non è stato ancora trovato un metodo del genere: scopo pertanto della nostra chiacchierata di oggi è descrivere pregi e difetti dei vari metodi lasciando a Voi la responsabilità di decidere quale tra i vari metodi di sort sia quello più adatto a particolari esigenze.

Per guidarVi in qualche modo nella decisione pubblichiamo una tabella in cui sono comparati i vari tipi di sort utilizzati per l'ordinamento di dati casuali.
Poiché alcuni metodi sono fortemente influenzati dal particolare ordine di partenza, nella tabella sono indicati anche i due casi estremi: il caso migliore (quando già i dati sono messi nel giusto ordine), il caso peggiore (i dati sono messi al contrario).
Non fate considerazioni impietose! Sappiamo che non avrebbe alcun senso eseguire il sort su dati già ordinati; questi tempi possono comunque dare una indicazione sui tempi necessari nel caso i dati siano, per caso, *parzialmente* ordinati.
I tempi pubblicati sono espressi in secondi e sono stati rilevati su un minicomputer e *non* su un Texas TI/99, essi, però, possono comunque costituire una valida indicazione relativa.

Nome del sort	Ordine dei dati	10	50	100	200	300
Bolle	migliore	<0.1	0.1	0.1	0.2	0.3
	casuale	0.1	2.6	10.8	43.0	95.7
	peggiore	0.2	4.2	16.7	67.3	151.2
Inserzione	migliore	<0.1	0.2	0.3	0.6	0.9
	casuale	0.1	1.5	5.7	22.9	49.9
	peggiore	0.1	2.8	11.3	44.9	100.4
Selezione	migliore	<0.1	1.2	4.4	17.2	38.4
	casuale	0.1	1.2	4.5	17.3	38.5
	peggiore	0.1	1.3	4.8	18.6	41.3
Indirizzi	migliore	1.4	1.5	1.5	1.7	1.8
	casuale	1.4	1.5	1.7	1.8	
	peggiore	1.4	1.5	1.7	1.8	
Shell	migliore	<0.1	0.3	0.6	1.6	2.7
	casuale	0.1	0.8	1.9	5.1	8.1
	peggiore	0.1	0.6	1.5	3.5	5.9

ANALISI DEI METODI

SORT A BOLLE

Nonostante il metodo di scambio attraverso coppie di dati sia quello forse più semplice e ovvio, questo sistema non è molto raccomandabile poichè i tempi medi per l'ordinamento di numeri casuali sono piuttosto alti e quelli per il caso peggiore sono addirittura disastrosi.
Comunque il tempo nel caso migliore è estremamente favorevole e si può dedurre che questo metodo sia valido nel caso di dati parzialmente ordinati.
I tempi medi sono alti e aumentano con il quadrato del numero dei dati, mentre il tempo minimo aumenta con l'aumentare del numero dei dati.

SORT A INSERZIONE

Nel sort a inserzione i dati vengono ordinati man mano che essi sono introdotti. I tempi medi e quelli massimi sono molto migliori di quelli del sort a bolle ma restano comunque abbastanza alti. Il tempo medio aumenta con il quadrato del numero dei dati, i tempi di caso migliore sono buoni e sono proporzionali al numero di dati.
È importante sottolineare che, dato che l'ordinamento viene fatto al momento dell'immissione dei dati da tastiera, diventa di fatto «invisibile» il tempo occorrente per il sort.

SORT DI SELEZIONE

In questo metodo viene scelto il termine più grande nella lista e scambiato con quello nella prima posizione.
Poi viene fatta, a partire dalla seconda posizione e così via. Se esistono *n* termini, saranno necessari soltanto *n* scambi ma sono, però, richieste numerose comparazioni.
Non ci sono notevoli differenze tra casi peggiori o migliori in quanto questo metodo è abbastanza insensibile all'ordine in cui compaiono i dati.

SORT A INDIRIZZI

Questo metodo è estremamente utile nel caso i dati da ordinare siano interi e di numero non eccessivamente elevato. Questa limitazione è dovuta al fatto che è richiesta una matrice per contenere le frequenze relative ai numeri da ordinare.
Il tempo richiesto è estremamente basso e non esistono differenze tra caso migliore e peggiore.
Poiché è richiesta una matrice supplementare l'impiego di memoria è più alto che negli altri tipi di sort.

SORT A SHELL

Questo metodo somiglia a quello a bolle in quanto anche qui vengono comparate e scambiate coppie di valori. Le differenze principali sono date dal fatto che la comparazione viene fatta con valori che sono a una certa distanza tra loro e, se uno scambio avviene, viene eseguita una nuova comparazione per decidere se sia necessario un nuovo spostamento.
I tempi sono abbastanza buoni e, cosa abbastanza curiosa, i tempi di caso peggiore sono più buoni che quelli di caso migliore.
Lo shell, pertanto, sarà particolarmente utile nel caso sia necessario ordinare dati mal disposti ed è in generale, il più veloce tra i metodi descritti.

CONCLUSIONI

Possiamo concludere dicendo che, in pratica, un metodo vale l'altro nel caso i dati da ordinare siano relativamente pochi; una certa cura va invece posta nella scelta del metodo quando i dati siano numerosi, la valutazione è lasciata, in questo caso, unicamente al programmatore che ben conosce le proprie particolari esigenze.
Auguriamo ai nostri amici buone vacanze e, al ritorno, di poter validamente impiegare queste tecniche per ordinare le agende telefoniche con nuovi indirizzi e nuovi nomi di persone simpatiche conosciute in vacanza.

PAOLO CIVARDI

SOFTWARE ASSEMBLER: PRINT

Come promesso, eccovi la presentazione di un programmino in assembler.
È un programma semplicissimo, ma sarà senz'altro molto utile ai neofiti, che potranno iniziare a prendere confidenza con la sintassi di questo linguaggio.
Come avrete potuto notare, i listati sono effettivamente due, abbiamo preferito separare le due versioni del programma (MINIMEMORY-EDITOR/ASSEMBLER), per ridurre al minimo le eventuali difficoltà di copiatura.
Ricordiamo che in questa sede sarà data solo una descrizione sommaria del programma, per quanto riguarda la spiegazione delle singole istruzioni, vi rimandiamo agli articoli sui prossimi numeri.

DESCRIZIONE PROGRAMMA

Iniziamo con la versione per EDITOR/ASSEMBLER.
La prima parte del programma, subito dopo il titolo, provvede alla definizione del nome (PRINT) e prepara i riferimenti per i sottoprogrammi che saranno utilizzati.
La seconda parte, prepara lo spazio riservato ai dati, nell'ordine: Indirizzo di ritorno, frase da videare, numero di colonna video, numero di riga video, lunghezza della frase, indirizzo del registro di stato, ed infine, l'indirizzo per la tastiera.
La terza ed ultima parte, non è che il programma effettivo. Per prima cosa viene effettuato il controllo sulla lunghezza della frase, se è <0>, il programma termina, altrimenti viene calcolata la posizione sul video in base alle coordinate <COL> e <RIG>, quindi viene trasferito il contenuto della riga <TESTO> a partire dalla posizione video calcolata in precedenza, per il numero di lettere specificate in <LUNG>, infine il programma, apparentemente si ferma, e vi permetterà di vedere su video la vostra frase. A questo punto, basta la pressione di un qualsiasi tasto, e il programma termina.
Le principali differenze con la MINIMEMORY, sono dovute sostanzialmente alla mancanza di istruzioni <DEF> e <REF> e alla limitazione a sole due lettere per i nomi simbolici.
L'istruzione <DEF>, viene simulata, andando a scrivere il nome direttamente nello spazio a lui dedicato, mentre le due <REF>, sono sostituite con due <EQU>.
Per quanto riguarda i simboli, è solo questione di comodità, basta fare attenzione a non scrivere due volte lo stesso simbolo.
Il resto del programma è identico all'altra versione, tranne le linee <COL> e <RIG> che sono condensate in un unica riga (CR), e la riga <LUNG> che diventa di due byte (LU).
Naturalmente è possibile modificare il numero della riga, il numero della colonna, ed anche la frase, non esistono limitazioni per quanto riguarda la lunghezza della frase, tranne quella del video (768 caratteri).
Bisogna però tener presente che il programma non effettua nessun controllo sulla validità dei dati eventualmente introdotti, sarà quindi opportuno porre molta attenzione onde evitare il blocco del sistema con annessa perdita del programma. Ricordate che i numeri di riga, di colonna e della lunghezza della frase, sono in esadecimale, che una riga video contiene al massimo 32 caratteri (i valori, dovranno quindi essere compresi tra <0> e <1F>), e che le righe sono 24 (valori tra <0> e <18>).
Nel caso in cui vogliate riempire tutto il video, i valori di riga e di colonna dovranno essere entrambi <0>, sempre per evitare il sopra citato blocco.
Ed ora, non ci resta che augurarvi Buon Lavoro!
SEBASTIANO TOMASELLO

```
*****************************************
*                                       *
*           < P R I N T >               *
*                                       *
*   VERSIONE PER   MINI  MEMORY         *
*                                       *
*****************************************


       AORG >7FE8
       TEXT 'PRINT '
       DATA IN

       AORG >701C
       DATA >7D74,>7FE8

       AORG >7D00
ST     EQU  >837C
TA     EQU  >8374
MW     EQU  >6028
KS     EQU  >6020
RT     BSS  2
TE     TEXT 'TI-99/4A NEW SOFT'
CR     DATA >080C
LU     DATA >1100

IN     MOV  R11,@RT
       CLR  R0
       MOVB @LU,R1
       ANDI R1,>FF00
       JEQ  FI
       MOVB @CR+1,R1
       JEQ  DO
       SWPB R1
LO     AI   R0,>20
       DEC  R1
       JNE  LO
DO     MOVB @CR,R1
       SWPB R1
       ANDI R1,>00FF
       A    R1,R0
       LI   R1,TE
       MOVB @LU,R2
       SWPB R2
       ANDI R2,>00FF
       BLWP @MW
       CLR  @TA
L2     BLWP @KS
       LI   R1,>FF00
       CB   @TA+1,R1
       JEQ  L2
FI     CLR  R0
       MOVB R0,@ST
       MOV  @RT,R11
       B    *R11
       END
```

```
*****************************************
*                                       *
*           < P R I N T >               *
*                                       *
*   VERSIONE PER EDITOR/ASSEMBLER       *
*                                       *
*****************************************


       DEF  PRINT
       REF  VMBW,KSCAN


STATUS EQU  >837C
TASTO  EQU  >8374
RET    BSS  2
TESTO  TEXT 'TI-99/4A NEW SOFT'
COL    BYTE >08
RIG    BYTE >0C
LUNG   BYTE >11

PRINT  MOV  R11,@RET
       CLR  R0
       MOVB @LUNG,R1
       ANDI R1,>FF00
       JEQ  FINE
       MOVB @RIG,R1
       JEQ  DOPO
       SWPB R1
LOOP   AI   R0,>20
       DEC  R1
       JNE  LOOP
DOPO   MOVB @COL,R1
       SWPB R1
       ANDI R1,>00FF
       A    R1,R0
       LI   R1,TESTO
       MOVB @LUNG,R2
       SWPB R2
       ANDI R2,>00FF
       BLWP @VMBW
       CLR  @TASTO
LOOP2  BLWP @KSCAN
       LI   R1,>FF00
       CB   @TASTO+1,R1
       JEQ  LOOP2
FINE   CLR  R0
       MOVB R0,@STATUS
       MOV  @RET,R11
       B    *R11
       END
```

TI 99 MARKET

Finalmente SOFTWARE di altissimo livello ed assolutamente inedito per il Vostro TI 99/4A.

A sole L. 30.000 cadauno (10% di sconto agli ABBONATI) *sono disponibili i seguenti programmi:*

CON SUPPORTO DISCO DA 5"

RADAR
2000 COWBOY
SNOOPY
QUADRO X
SCOPA

LA DESCRIZIONE E' A PAG. 26

CON SUPPORTO CASSETTA MAGNETICA:

TOMBOLA
+ GIROTONDO
ELESIUS
+ 1000 NOMI
TI 99 CALC

LA DESCRIZIONE E' A PAG. 26

PER ORDINARE IN CONTRASSEGNO
(+ spese di spedizione):

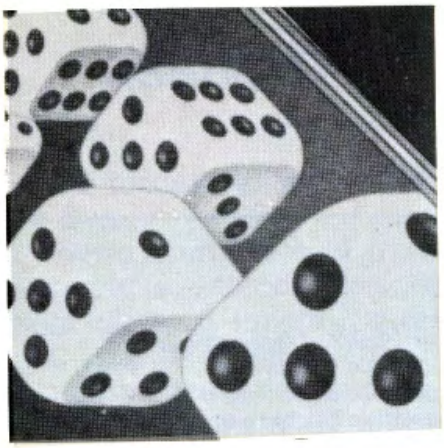
TI 99 MARKET
Via Stefano Jacini, 4 - 20121 MILANO

YATZEE

Se qualcuno di voi non ha mai giocato a POKER, questa è una buona occasione per incominciare.
YATZEE è un gioco basato sul lancio di cinque dadi, in base al punteggio ottenuto, possiamo scegliere tra diversi tipi di giocata. Per motivi di spazio non possiamo spiegarvi le regole del POKER, interpellate qualche vostro amico o parente, le sapranno certamente.

IL GIOCO

Le regole del gioco sono più o meno uguali a quelle del POKER, fatta eccezione per il fatto che al posto delle carte, troveremo i dadi.
Abbiamo a nostra disposizione tre lanci per tentare di ottenere il maggior punteggio. Ad ogni lancio possiamo fermare uno o più dadi, premendo il tasto numerico (1, 2, 3...) corrispondente al valore del dado da noi scelto. Ovviamente i dadi fermati non saranno lanciati nuovamente.
Per lanciare i dadi useremo la barra spaziatrice.
Per i più distratti sono previsti dei tasti per cancellare eventuali scelte sbagliate.
Il tasto - 0 - «zero» annulla le scelte sino a quel momento effettuate, oppure - R - seguito dal valore del dado che vogliamo cambiare.



Quest'ultima soluzione non annulla tutte le scelte, ma solamente quella da noi desiderata.
Il tasto - A - ci permette di fermare tutti i dadi presenti in quel momento.
In base al risultato ottenuto, andremo ad indicare con la freccia di selezione il tipo di puntata per noi più appropriata, poi sposteremo la freccia di una casella a destra e premeremo - ENTER -.
Il punteggio verrà visualizzato e potremo ricominciare a tirare i dadi.
La freccia di selezione si muove utilizzando i tasti:
- S - muove la freccia di una casella a sinistra
- D - muove la freccia di una casella a destra
- E - muove la freccia di una casella in alto
- X - muove la freccia di una casella in basso
Dobbiamo tener presente che ogni giocata è unica, si può fare solamente una volta.
Quando abbiamo terminato le giocate a nostra disposizione il gioco è terminato, vince chi riesce a fare più punti.

EZIO MONTINI

DESCRIZIONE DEL PROGRAMMA - YATZEE -

Linee	Descrizione della funzione
100 - 250	Inizializzazione delle variabili, caricamento dati grafici
260 - 340	Input dei dati (n. giocatori, n. partite, ecc.)
350 - 430	Scelta del giocatore che incomincia per primo
440 - 530	RND dadi e visualizzazione degli stessi
540 - 800	Costruzione del quadrante di gioco e spostamento della freccia per la selezione delle puntate
810 - 1320	Printaggio dei BONUS e messaggi vari
1330 - 1550	Dati numerici e di stringa

VARIABILI NUMERICHE

A, B, G, H, F, Y	Variabili di comodo
E	Variabile contenente il valore di ogni dado tirato
E1	Contiene il numero di caselle di spostamento orizzontale
I	Contatore numerico ciclo FOR-NEXT per caricare i codici caratteri
L1	Variabile contenente il codice di ogni dado tirato
PT	contiene la sommatoria dei punti dei dadi tirati
Q5	Variabile contenente il numero dei giocatori (max. 6)
R	Se POKER è stato fatto, contiene il valore del punteggio del POKER
R1	Contiene il numero di lanci dei dadi eseguiti
SU	Contiene il codice ASCII del dado da visualizzare
T	Variabile di stato della CAL KEY
V, V3	Variabili di ritorno della CALL KEY
V2	Contiene il valore di V-48, indicante il valore del dado che si desidera tenere
X	Contiene il codice ASCII dei caratteri grafici definiti
YT	Indica quale tipo di BONUS si deve assegnare
Z	Contiene il numero di partite che si desidera giocare (max. 3)

VARIABILI MATRICIALI NUMERICHE

D (5)	Matrice contenente il valore dei dadi da visualizzare
DV (6)	Matrice contenente la variabile casuale da assegnare ad ogni dado
GT (6)	Matrice contenente il punteggio aggiornato di ogni giocatore
S (6, 3, 16) ..	Matrice a tre dimensioni contenente il resoconto di tutte le puntate e il punteggio che ogni giocatore ha ottenuto in ognuna di esse
Y5 (5)	Matrice contenente i BONUS assegnati ai giocatori

VARIABILI MATRICIALI DI STRINGA

Q$ (6)	Matrice contenente i nomi dei giocatori

VARIABILI DI SIRINGA

C$, C1$	Contengono la descrizione dei BONUS visualizzata alla riga 24
M$	Contiene la stringa dei caratteri da definire

EQUO CANONE

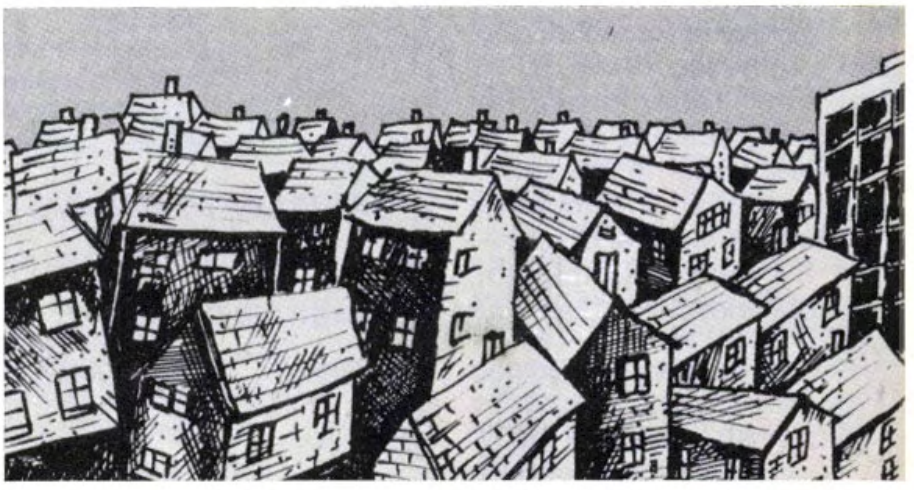
Siete proprietari o inquilini di un appartamento o di una villa?
Bene, questo programma vi permetterà di conoscere in pochi minuti se il canone di affitto che pagate o riscuotete è quello esatto.
Non dovete fare altro che rispondere con la massima precisione alle domande che il computer vi pone ed in breve avrete la cifra mensile esatta dovuta per il vostro appartamento.
Nel programma sono già calcolati fino all'anno 1983 tutti gli aumenti ISTAT che hanno avuto luogo dall'uscita della legge relativa.
Naturalmente ogni anno successivo, in genere a fine luglio, occorrerà aggiungere il nuovo aumento che corrisponde al 75% dell'aumento del costo della vita e che tutti i giornali pubblicano nello stesso periodo.
A questo scopo sono state lasciate libere alcune righe nel programma. Se per esempio nell'anno '84 l'aumento fosse dell'11,9% dovrete sostituire al PRINT nella riga 2670 la seguente formula:
CAN = CAN + (CAN × 11,9/100)
ed avrete l'Equo Canone aggiornato.
LEVIO PEROTTI

ANALISI DEL LISTATO

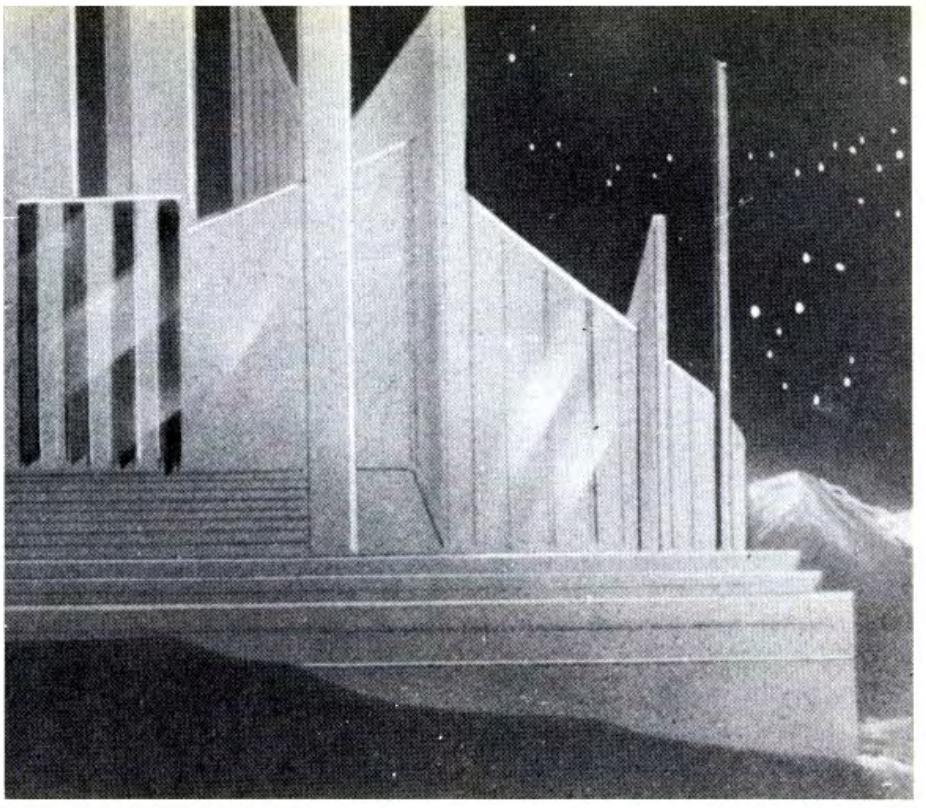
100 - 140	Titolo
150 - 200	Richiede e calcola la superficie dell'appartamento
250 - 380	Richiede e calcola la superficie autorimessa
390 - 500	Richiede e calcola la superficie posto macchina
510 - 620	Richiede e calcola la superficie balconi e terrazze
630 - 740	Richiede e calcola la superficie giardino privato
750 - 890	Richiede e calcola la superficie condominiale
900 - 1020	Richiede e calcola la superficie con altezza inferiore a mt. 1,70
1030 - 1080	Richiede e calcola l'anno di costruzione dell'immobile
1090 - 1220	Richiede e calcola la posizione geografica
1230 - 1500	Richiede e calcola la categoria catastale
1770 - 2080	Richiede e calcola l'ubicazione dell'immobile
2090 - 2240	Richiede e calcola a che piano si trova l'immobile
2250 - 2390	Richiede e calcola la vetustà dell'immobile
2610	Calcola l'equo canone base
2620 - 2660	Calcola gli aumenti ISTAT a tutto il 30 luglio '84
2670 - 2700	Righe libere per i successivi aggiornamenti ISTAT
2710 - 2730	Presenta il risultato dei calcoli
2740 - 2780	Istruzioni per immettere i successivi aumenti ISTAT

VARIABILI UTILIZZATE

AUT	Superfice autorimessa
BAL	Superfice balconi e terrazzi
BAS	Superfice con altezza inferiore a mt. 1,70
CAN	Equo canone risultante
ANNO	Anno di costruzione
CO	Superfice condominiale
DAT	Anno corrente
EQ	Equo canone base
GIA	Superfice del giardino
MIL	Quota millesimale
N	Vetustà immobile
PM	Superfice posto macchina
SUP	Superfice dell'appartamento
Z, X	Variabili temporanee di comodo



ASSEDIO



Siamo tornati nell'antico Medio Evo.
Il nostro compito è di fronteggiare un attacco barbaro dalla cima di un'alta muraglia. L'unica arma di cui disponiamo è una riserva praticamente illimitata di pietre pesantissime da lanciare contro i barbari in modo da farli cadere.
Ogni volta che getteremo una pietra, però, tutti i barbari faranno un balzo verso l'alto, anticipando così l'invasione del castello.
Quando uno dei barbari giungerà in cima verrete catturati.
A questo punto se il capo dei barbari vorrà essere crudele, la vostra fine sarà pure imminente, altrimenti... insomma lo vedrete giocando, non vogliamo togliervi il gusto di scoprirlo da voi.
Per muoversi usate i tasti «S» e «D» e la barra spazio per lanciare i massi.

DANIELE CATALFAMO
(MMG)

LISTATO

100 - 320	Inizializzazione, definizione caratteri e colori
330	Richiama la subroutine dell'intestazione
370 - 390	Inizializza a 24 il numero di riga di tutti gli assaltatori
420 - 600	Ciclo principale
440 - 490	Legge la tastiera e se è stato premuto un tasto controlla se deve saltare alla subroutine di lancio del masso oppure muovere l'omino
500	Sceglie a caso uno degli assalitori
510 - 610	Controlla se l'assalitore è già sullo schermo quindi lo stampa nelle nuove coordinate dopo averlo cancellato (se già sullo schermo)
570	Controlla se l'assalitore è arrivato in cima
620 - 760	Subroutine che controlla se sono stati colpiti degli assalitori quando è stato lanciato il masso, in tal caso ne reinizializza le coordinate e attribuisce il punteggio
770 - 900	Subroutine che visualizza la caduta del masso
910 - 960	Quando uno degli assalitori è arrivato in cima viene richiamata questa subroutine che fa dirigere l'assalitore verso l'omino
1010	Decide se finire subito o concedere la possibilità di un'altra partita
1050 - 1240	Chiede su quale numero scommetti, quindi estrae un numero a caso e stampa il dado con il risultato della scommessa
1250 - 1300	Chiede se vuoi giocare ancora oppure finire
1350 - 1400	Stampa la pagina introduttiva

VARIABILI

P (15,1)	Matrice contenente le coordinate degli assalitori
X	Coordinata in cui si trova l'omino
K, S	Usate nelle CALL KEY
I	Usata sia per cicli FOR-NEXT che sia come indice per individuare il nemico da far salire
CAR	Incrementa o no il codice ASCII del carattere dell'assalitore in modo che venga stampato una volta con un braccio alzato e una volta con l'altro braccio
SC	Contiene il punteggio
N	Contiene la scommessa del giocatore
D	Contiene il numero estratto
V, Q	Usate per i cicli FOR-NEXT

BIORITMI

I bioritmi descrivono il flusso biologico che accompagna l'individuo dalla nascita. Tre sono le componenti che ne determinano il divenire:
— il ciclo fisico, della durata di 23 giorni;
— il ciclo emotivo, della durata di 28 giorni;
— il ciclo intellettivo, della durata di 33 giorni.
Come si può notare questi cicli si possono definire compiutamente visualizzandone il flusso nell'arco di circa trenta giorni, quindi con una progressione mensile.
Ogni ciclo è diviso in due fasi: una positiva durante la quale le cellule del corpo sono cariche di energia e l'individuo è al meglio della forma, ed una negativa che è quella di ricarica, utilizzata dal nostro corpo per recuperare le energie spese.
Critici sono detti i giorni di passaggio da una fase positiva ad una negativa e viceversa.

IL PROGRAMMA

Il programma scritto in **TI BASIC gira anche in EXTENDED.**
Per prima cosa vi verrà chiesto il vostro nome; questo perché è possibile memorizzare all'interno del programma nomi e date di nascita inserendoli a partire dalla linea 1370 in questo modo: n. linea DATA NOME, GIORNO, MESE, ANNO.
Il nome dev'essere scritto in lettere maiuscole.
Riguardo alle date di nascita, il programma accetta:
— giorno coerente al mese;
— mese in cifre o in lettere;
— anno di quattro cifre (con la facoltà di omettere le prime due nel caso si tratti del XX° secolo).
L'ultima linea di DATA dev'essere: n. linea DATA ÷, 1, 1, 1. Pena lo scoppio del programma!!!
Per i nomi non memorizzati vi verrà richiesta la relativa data di nascita.
Successivamente dovrete indicare il periodo, e precisamente mese ed anno, relativamente al quale desiderate analizzare i bioritmi.
Subito sarà visualizzata la tabella che interessa, e su di essa verranno tracciati i diagrammi dei tre cicli individuati dai simboli sotto indicati:
* per il ciclo fisico;
∂ per il ciclo emotivo;
÷ per il ciclo intellettivo.
Fase positiva è considerata quella durante la quale il diagramma percorre la fascia superiore rispetto all'asse, mentre l'inferiore va valutata come zona di ricarica.

RICCARDO ROSSI

CARATTERISTICHE DEL PROGRAMMA

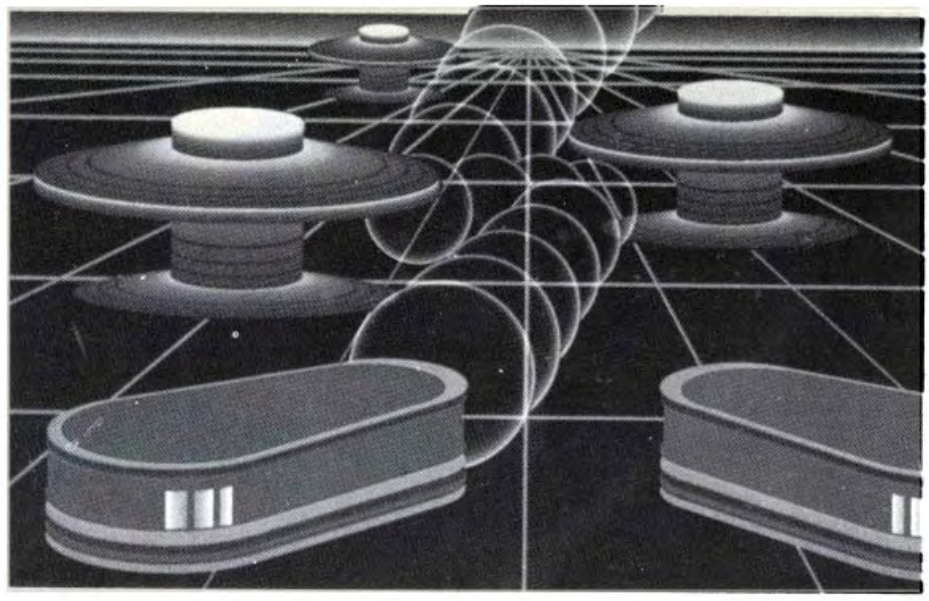
LISTATO

linee	descrizione della funzione
180 - 190	Definizione caratteri
200 - 220	Caricamento tabelle
250	Definizione di pigreco
260	Eliminazione minuscole dall'input
300 - 340	Input nome e verifica se è presente in memoria
350 - 470	Input data di nascita e verifica della correttezza
500 - 650	Input data dell'analisi e verifica
660 - 800	Stampa tabella mese
810 - 890	Stampa bioritmi
900 - 1040	Testo per la continuazione del programma
1050 - 1080	Tracciamento sinusoidi dei cicli bioritmici
1090 - 1220	Verifica della correttezza del mese
1230 - 1280	Calcolo dei giorni dall'anno 0
1290 - 1370	Controllo dell'anno bisestile
1370	Inizio linee DATA contenenti: NOME, GIORNO DI NASCITA, MESE DI NASCITA, ANNO DI NASCITA

LISTA DELLE PRINCIPALI VARIABILI

MM$	Tabella nome mesi
GR (12) ...	Tabella giorni-mese
K, S	Lettura tastiera
NOMES ...	Nome dell'analizzato
PI	Pigreco
A$	Stringa per il confronto del nome
M	Mese
GG	Giorno
M$	Mese (solo in input)
AA	Anno
GN	Giorno di nascita
MN	Mese di nascita
AN	Anno di nascita
DN	Numero giorni dall'anno 0 alla nascita
DD	Numero giorni dall'anno 0 al mese dell'analisi
DIF	Numero giorni dalla nascita al mese dell'analisi
P	Durata del ciclo
S	Codice carattere del ciclo

THE WALL



È un riadattamento di uno dei più vecchi video-games da bar, che conserva sempre il suo vecchio fascino.
È possibile giocare sia con la tastiera che con il joystick.
Come spiegato abbondantemente nei titoli, il vostro compito è quello di abbattere completamente un muro formato da quattro file di mattoni colorati, lanciandogli contro una palla.
Ogni mattone abbattuto vi permette di raggiungere il successivo che naturalmente vi da un punteggio maggiore.
Quando riuscirete a praticare un passaggio nel muro e a infilarci dentro la pallina, questa continuerà a rimbalzare disperatamente tra la parete di fondo e le ultime file di mattoni, finché non riuscirà nuovamente ad uscire avendovi fatto guadagnare un mare di punti.
La palla non viene perduta finchè riuscite a rinviarla contro il muro e dopo un po' di pratica riuscirete a raggiungere il punteggio più alto (32219 punti) guadagnando così il diritto ad un nuovo muro da abbattere sommando i nuovi punti al vecchio punteggio.
Il punteggio record viene mantenuto in memoria dal computer e presentato ad ogni fine partita.

Nonostante una certa lentezza della palla e della racchetta dovuta alla bassa velocità del T.I. Basic, Vi assicuro che il gioco è abbastanza emozionante.
Ah, qualche consiglio: per spuntarla su quella maledetta pallina cercate di mantenere la racchetta al centro del campo e, quando lanciate in diagonale, non restate fermi a guardare la palla, ma seguite la traiettoria prevedendo che la pallina tornerà giù con un angolo opposto a quello di battuta.
Non disperate all'inizio se la palla sembra scapparvi da tutte le parti, migliorerete con la pratica e... si vocifera in redazione che il Direttore metterà in palio un abbonamento a vita per chi dimostrerà (computer acceso) di aver raggiunto il 10° quadro.

LEVIO PEROTTI

LINEE	*Descrizione della funzione*
100 - 220	Titolo
230 - 300	Istruzioni
310 - 430	Scelta tastiera o joystick
440 - 610	Istruzioni
620 - 720	Definizione caratteri
730 - 810	Assegnazione colori
820 - 1000	Azzeramenti e inizializzazione
1010 - 1180	Routine di scrittura sul video
1190 - 1260	Disegno del campo di gioco
1270 - 1280	Routine apparizione pallina
1290 - 1450	Conteggio delle palline
1460 - 3810	Movimento e controllo della pallina
3820 - 3930	Motivetto di fine partita
3940 - 4020	Presentazione punteggio e record partite precedenti
4030 - 4270	Richiesta di nuova partita

VARIABILI UTILIZZATE	
A, B	Definiscono la posizione della pallina
E	Colore del quadro
F	Verifica la coincidenza della pallina con la racchetta o con i mattoni
L	Punteggio realizzato
W	Massimo punteggio realizzabile con un quadro
U, D, C, M, DM, CM	Cifre che definiscono il punteggio
UR, DR, CR, MR, DMR, CMR	Cifre che definiscono il record raggiunto
N, O, P	Cifre che definiscono il numero delle palline
T	Accerta se si gioca con il joystick o con la tastiera
R, Z, X, Y	Variabili temporanee di comodo

PERCHE' NON FUNZIONA?

...Lodevole la Vostra pubblicazione sia dal lato grafico che da quello nozionistico per un neofita di computer quale io sono.
Trovo poi le cassette veramente superbe perché uniscono il puro divertimento a nozioni più specifiche ed utili anche ad una famiglia. Mio figlio è rimasto colpito in modo entusiastico dell'ARITMETICA VELOCE e, facendo la terza elementare, penso possa essergli di una certa utilità.
Ho però dovuto arrendermi di fronte ad un mistero circa il gioco *dello SLALOM SPECIALE*. Vengo a Voi per chiederVi, gentilmente come posso ovviare a detto mistero. Quando il gioco inizia con la scritta del titolo, sullo schermo appaiono i vari livelli di difficoltà che si possono *scegliere (1-20, PERCORSO)* ma, qui avviene una cosa strana che non mi ha ancora consentito di vedere questo gioco. Una volta scelto ed impostato il numero della difficoltà (2, per esempio) dò l'ENTER per vedere il gioco ma il computer mi risponde: INPUT ERROR IN 680, TRY AGAIN. Ho provato, ho riprovato con calma ma ho sempre ottenuto lo stesso cattivo risultato. Perché?

ZAPPA GIANCARLO

In effetti dobbiamo fare «mea culpa», in quanto per una svista, dobbiamo dire piuttosto banale, il programma «Slalom gigante», non è che presenti degli errori nel listato, bensì manca delle informazioni relative alle INPUT da dare. E così veniamo alla spiegazione. In effetti quando il TI 99 vi chiede di scegliere la Difficoltà impostando un numero da 1 a 20, poi vi chiede anche di scegliere il tipo di PERCORSO.
Qui c'è la svista. Manca infatti, vicino alla parola Percorso la parentesi contenente i numeri 1-20, che sono i vari percorsi che si possono scegliere. Quindi, quando digitate un numero da 1 a 20 per la Difficoltà, non dimenticate di farlo anche per il Percorso. Ovviamente i due numeri devono essere separati da una virgola.
A questo proposito vi consigliamo di leggere, guarda caso, l'articolo sulle istruzioni di I/O a pagina 3.

VENDO COMPRO CAMBIO SCAMBIO

VENDO

Vendo joistick original Texas Instruments ancora nel loro imballo che non è mai stato aperto a Lit. 45.000 spese di spedizione comprese. Ferri Stefano - Via Toscanini 8 - Castelmassa (RO) - Tel. 0425-81101.

Vendo **TI 99** + Cassetta basic + Cavi + registratore IRRADIO + altri programmi L. 400.000 trattabili. Andrea Guelpa - Via Leoncavallo 45/D - Torino - Tel. (011)288040 ore (14-22).

Affare!! Vendo **TI 99/4A,** con registratore, cavetto, manuale, scacchi, modulatore TV, libro dedicato, il tutto poco usato a L. 250.000. Nicolai Alfonso - Fizzonasco (MI) - Via Cafiero 1 - Tel. 02/8266935 oppure 02/907230.

Vendo, causa passaggio di sistema, **TI-99/4A** con: cavetto per registratore - modulo Miny Memory + ling. Assembler + manuale Assembler (500 pag.) - modulo TI Emulator II (trans. dati) - modulo Parsec (gioco velocissimo, tipo bar) - modulo Soccer (calcio) - Speech Synthesizer (dà la voce al computer) - 4 cassette con software Basic - 5 libri per TI-99 - Joystick Spectravideo (eccezionale) + interfaccia per TI-99. Valore nuovo L. 1.180.000. Il tutto in ottime condizioni, con imballi e documentazione originali. In blocco vendo a L. 785.000. Vendo anche separatamente, prezzo da concordare. Scrivere o telefonare a: Bartoli Paolo - Via Grigna 2 - 20052 Monza - Tel. 039/324790.

CAMBIO

Cambio **Car Wars Zero Zap** Video Games 1 con altri giochi su cassetta o su cartuccia solo zona di Milano. Telefonare a: Staffiero Gianandrea 02/681104 - ore serali.

Cartuccia **"Parsec"** con qualsiasi altra cartuccia dello stesso valore o con Extended Basic (eventualmente aggiungo una somma). Sorce Giuseppe - C.so Giovanni Agnelli 32 - Settimo Torinese (TO) - Tel. 8006536.

Per TI-99/4A scambio **Speech Synthesizer** in cambio cerco modulo minimemory. Dispongo anche dello Speech Editor. Telefonare o scrivere a Florio Michele 075/79472 (ore ufficio) 075/798832 (ore pasti fino ore 20). Via Settembrini 8/A5 - 06070 S. Mariano (PG).

CERCO

Cerco **Modulo Extended Basic** per TI-99/4A. Acetto qualsiasi ragionevole prezzo. Ippolito Corsico - Tel. 02/4409757 ore serali.

Cerco **Modulo Extended Basic** per TI-99/4A telefonare o scrivere a: Alessandro Tornetta - C.so Vercelli 254 - 10155 Torino - Tel. 011/263463 ore pasti.

Cerco e compro possibilmente a buon prezzo e in buone condizioni, il **Modulo Extended Basic** per TI-99/4A. Loris Delmastro - Via Novara 123 - Romagnano Sesia (NO) - Tel. 0163/832888 ore serali.

Per **TI-99/4A** cerco, solo se affare, **Minimemory** (con relativo manuale) + SSS scacchi + Joystick (solo originali). Telefonare 0765/24430 o 24048 chiedendo di Piero, oppure scrivere a Cipriani Piero - Via Nardi 14 - 02047 Poggio Mirteto (RI).

COMPRO

Compro a buon prezzo, preferibilmente zona Palermo Modulo extended basic per **TI-99/4A**. Piero Romeo - Via Nairobi 39 - 90129 Palermo - Tel. 091/426755 (ore pasti).

Compro modulo Extended Basic per **TI-99/4A** con/senza manuale - Telefonare ore pasti o serale: 011/381661 - Adalberto Bonardi - Torino.

Compro modulo Extended Basic per **TI-99/4A. Prezzo da convenirsi. Telefonare dopo le 21 allo 080/420161. Oppure scrivere a: Roberto Murri - Via de Laurentis 13/D - 70124 Bari.**

Compro Texas Instruments TI-99/4A a L. 240.000. Chi è interessato al suddetto annuncio è pregato di telefonare allo 06/7853011 (Roma).

Compro **Texas TI-994A** a prezzo ragionevole offro lire 150.000. Vendo Drive Disk. Shugart. A lire 800.900 + 2 floppy in omaggio (trattabile). Telefonare a Bertocchi Carlo Alberto - Tel. 4585420 - Via Nazario Sauro 12 - Cesano Boscone 20090 (MI).

Compro per **TI-99/4A** modulo Minimemory + cassetta con programmi "Lines" e "Line by line Assembler" a meno da 735.000 lire. Compro anche disk drive esterno, Editor Assembler e TI Emulator II. Telefonare o scrivere a Giorgio Longo - Via Volturno 80 - Brugherio (MI) - Tel. 039/878185. Sono in casa dalle 14 alle 17 escluso il giovedì.

Compro modulo Mini-Memory per **Texas TI-99/4A**, completo dei manuali d'uso, della cassetta con i programmi "Assembler line by line" e "Lines" e possibilmente con la propria confezione originale. Mettersi in contatto con Tirabassi Cesare - Via Batà, 6 - 62100 Macerata o telefonare ore pasti allo 0733/48306.

Compro **modulo Minimemory** + cassetta line by line Assembler per TI-99/4A. Possibilmente anche manual editor Assembler. Mollo Enrico - Via Martiri n. 23/A - 12014 Demonte (CN).

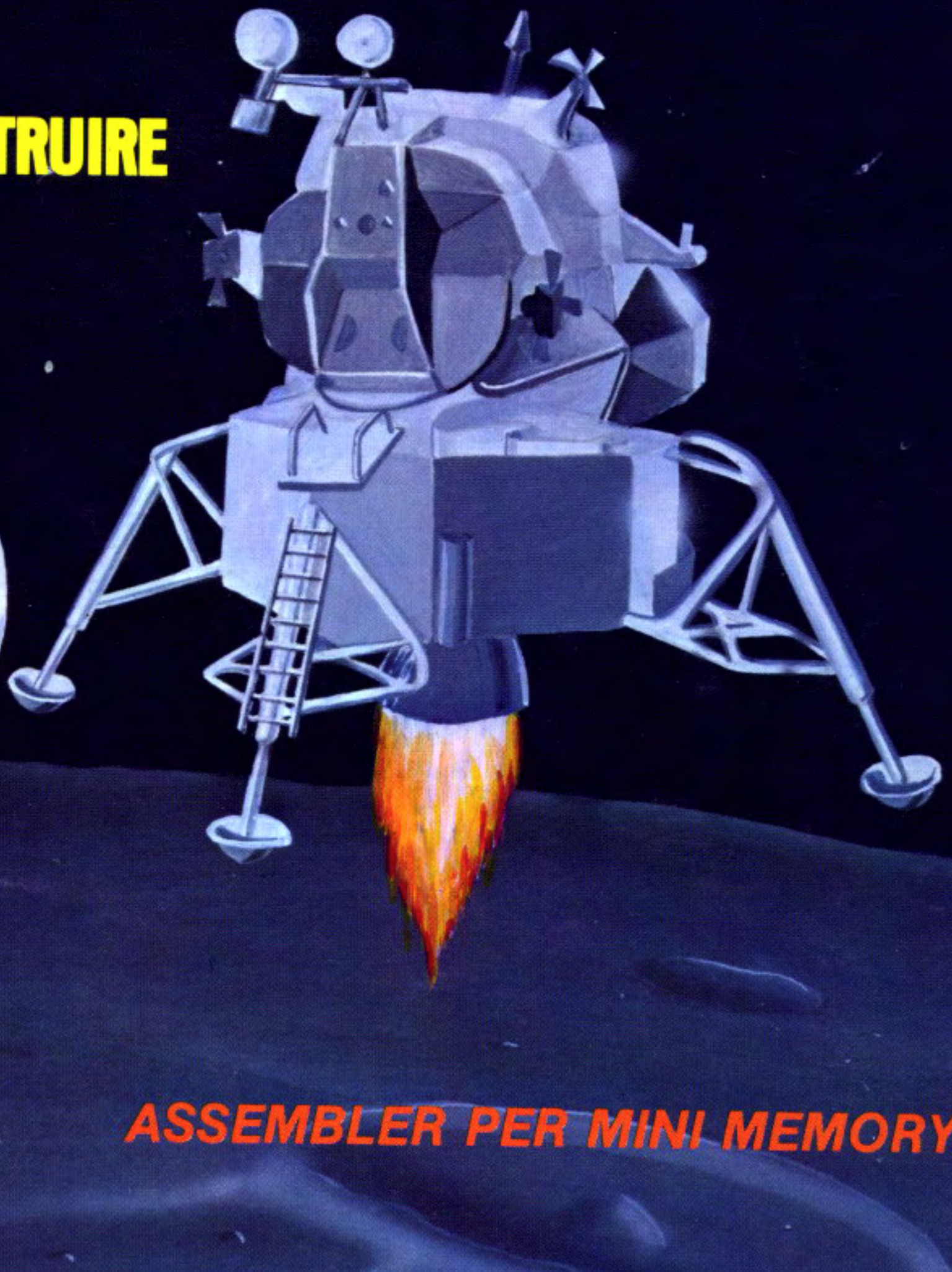
Compro solo se occasione **Minimemory** completa di manuale e cassetta; emulatore, sintetizzatore per TI-99/4A. Preferirei essere contatto da amici sardi. Tel. 070/932461 - Ignazio Fanni - Via Roma 41 - 09039 Villacidro (CA).

TI 99 NEWSOFT

NOVEMBRE 1984 L. 10.000

EDISOFT

IMPARIAMO A COSTRUIRE UN ARCADE GAME: Gli Sfondi mobili



ASSEMBLER PER MINI MEMORY

BASIC: Gli operatori logici

IN SOFWARE & SOFTWARE
Grand Prix - LANDER - Coyote - Software Mini Memory ...

SOMMARIO

□ **INUTILITIES** pag. 2

□ **GLI OPERATORI LOGICI**
Come simulare gli operatori logici in TI BASIC pag. 3-4

□ **IMPARIAMO A COSTRUIRE UN ARCADE GAME**
Questa volta parliamo degli Sfondi Mobili che rendono un videogame più avvincente pag. 5-7

□ **QUASI UN ARCADE**
E' una routine in Extended Basic che con poche modifiche diventa un buon videogioco pag. 8

□ **UNA ROUTINE DI COLLOQUIO**
E' un programma che consente di utilizzare i programmi che abbiamo pubblicato nel numero scorso: CHAR DEF e Ti Text File pag. 9

□ **MINI MEMORY**
Una nuova rubrica dedicata al modulo SSS. Questa volta parliamo della Mappa RAM pag. 10-11

□ **SOFTWARE MINI MEMORY**
Per ridefinire velocemente i caratteri dei vostri programmi in questo numero: LOADER CARATTERI e CARATTERI LINE pag. 12-13

□ **EXTENDED BASIC**
Ancora una puntata sugli Sprites. Questa volta li muoviamo con i JOYSTIK pag. 14

□ **TI 99/4A**
Una rubrica per rispondere alle vostre domande pag. 15-16

□ **GUARDIA E LADRI**
Un giochino simpatico in BASIC. Dovete cercare di "rubare" il più possibile pag. 16-17

□ **LANDER**
È il tema di copertina. Ci vuole molta abilità e prontezza di riflessi. Auguri! pag. 18

□ **SPACE INVADERS**
Gira in Extended ed è velocissimo, quasi quanto il modulo SSS pag. 19

□ **STATISTICA**
A cosa serve la statistica? Vi forniamo alcune risposte pag. 20-21

□ **INVENZIONI** pag. 21

□ **COYOTE**
Un bel gioco in Extended per due persone pag. 22

□ **GRAND PRIX**
Una gara automobilistica in Basic. Buona grafica pag. 23-24

□ **TI 99 MARKET** pag. 25-26

□ **PICCOLI ANNUNCI** pag. 28-29

□ **LA VIGNETTA**
È la prima vignetta dei nostri lettori. Vince un abbonamento omaggio. pag. 30

□ **COLLABORATE CON NOI**
Se avete dei programmi che volete vedere pubblicati su TI 99 NEWSOFT, mandateceli accompagnandoli con questo modulo pag. 31-32

TI 99 NEWSOFT

ANNO 1 - Numero tre
NOVEMBRE 1984
MENSILE
Direttore responsabile:
ANTONIO C. LOSITO
Coordinatore Editoriale:
VINICIO CIVININI
Art Director:
BEPPE RE-FRASCHINI
Illustrazioni:
MARIA TERESA COCOZZA
Segreteria:
EMANUELA CHIAPELLO
CRISTINA FRISON
Redazione:
FIORENZA AURIEMMA
DARIO BAREZZI
Collaboratori:
SERGIO BORSANI,
GIANPAOLO BOTTIN,
DANIELE CATALFAMO,
PAOLO CIVARDI,
STEFANO DELLI PONTI,
FRANCESCO MACALI,
MAURO MATTIAZZI,
EZIO MONTINI,
LEVIO PEROTTI,
RICCARDO ROSSI,
MARCO SQUINTANI,
ALBERTO STRAFILE,
SEBASTIANO TOMASELLO,

TI 99 NEWSOFT è una pubblicazione della **NEWSOFT snc**, via Stefano Jacini 4 - 20121 MI
Tel. 02/807464 - Telex 324284
REGISTRAZIONE AL TRIBUNALE DI MILANO N. 323 DEL 7.7.84

© COPYRIGHT NEWSOFT snc
TUTTI I DIRITTI RISERVATI
Manoscritti e foto originali, anche se non pubblicati, non si restituiscono ed è vietata la riproduzione anche parziale di testi, fotografie e listati.

Pubblicità: AV STUDIO
Via Giuseppe Meda, 23
Tel. 02/8357155
Abbonamenti: l'abbonamento annuo a 11 numeri di *TI 99 NEWSOFT* costa L. 90.000 per l'Italia; L. 150.000 per l'Europa (spedizione aerea); L. 200.000 per tutti gli altri paesi. L'importo, per assegno non trasferibile o per vaglia postale, va inviato a NEWSOFT snc, via S. Jacini 4 - 20121 Milano.
Stampa: CGB, Albegno di Treviolo (BG)
Concessionaria per la distribuzione: EUROSTAMPA Srl - C.so Vittorio Emanuele 111 - 10123 Torino - Tel. 011/538166-7

L'editore non si assume responsabilità alcuna sul contenuto degli articoli e/o software inviati. Per gli articoli firmati e/o siglati da collaboratori esterni, la redazione si assume la responsabilità prevista dalle leggi sulla stampa.

Tutti possono collaborare a TI 99 NEWSOFT, sia con articoli che con programmi. Scriveteci o telefonateci in redazione per avere tutti i ragguagli.

INUTILITIES

Saper scegliere un dono è un arte, diceva qualcuno.

E in tempi come questi un regalo utile risulta ancora più gradito.

Molti di voi l'anno scorso hanno brillantemente risolto il problema acquistando un computer, magari proprio il nostro TI 99/4 A.

Un regalo utile, senza dubbio, che potrete rendere ancor più completo aggiungendoci quest'anno un abbonamento alla nostra rivista TI 99 Newsoft; undici numeri ricchi di idee, suggerimenti e possibilità per sfruttare le capacità del vostro computer, per il gioco, il lavoro, lo studio, il tempo libero; per fare del computer uno strumento di lavoro quotidiano.

Un amico in più al vostro fianco.

Parlatene a Babbo Natale, anche lui ne sarà entusiasta.

GLI OPERATORI LOGICI

Come ben saprete, il TI BASIC è sprovvisto degli operatori logici AND, OR e XOR, che si rivelano molto utili al programmatore, in quanto lo liberano dalla necessità di dover impiegare confronti multipli su più righe.
Il problema può essere risolto abbastanza facilmente, ma, prima di spiegarvi come, parliamo di queste nuove paroline.
Un'espressione logica con AND è vera se entrambe le relazioni sono vere. Esempio:
10 IF 1<2 AND 4>0 THEN 100
L'interprete salterà alla linea 100.
Un'espressione logica con OR è vera se almeno una delle relazioni (ma anche entrambe) risulta verificata. Esempio:
10 IF 2<1 OR 5< >5 THEN 100
Il computer non salterà alla linea 100 poichè i confronti sono entrambi falsi.
10 IF 2>1 OR 4=3 THEN 100
Il computer salterà alla linea 100 in quanto il primo confronto è vero.
Con l'operatore XOR soltanto una delle relazioni deve essere vera perché sia vera l'espressione (XOR sta per exclusive OR). Esempio:
10 IF 3>0 XOR 5><3 THEN 100
Il computer non salterà alla linea 100 poichè entrambi i confronti sono verificati.

Per ulteriori chiarimenti sui tre operatori fare riferimento alla figura 1.

AND = *; OR = +; XOR = —.

Fig. 1 - (A e B sono relazioni)

AND

A	B	A AND B
falso	falso	falso
falso	vero	falso
vero	falso	falso
vero	vero	vero

OR

A	B	A OR B
falso	falso	falso
falso	vero	vero
vero	falso	vero
vero	vero	vero

XOR

A	B	A XOR B
falso	falso	falso
falso	vero	vero
vero	falso	vero
vero	vero	falso

L'interprete BASIC assegna il valore 0 alle relazioni false e —1 a quelle vere. Ciò può essere sperimentato dacilmente: il comando PRINT 1>2 scriverà 0 sullo schermo, mentre il comando PRINT 2 = 2 scriverà —1.
Tenendo conto che con l'istruzione IF... THEN... l'interprete salterà alla linea indicata dopo il THEN quando il valore dell'espressione di seguito all'IF è diverso da 0, si può arrivare a simulare i tre operatori logici AND, OR, XOR con i tre operatori aritmetici *, +, — rispettivamente.
Bisogna però ricordare che ogni confronto deve essere messo tra parentesi perché il presente metodo funzioni.
Nella figura 2 sono riportati alcuni esempi sicuramente delucidativi.
Questo sistema risulta efficace anche usando più di due confronti per istruzione IF... THEN, ma non sempre quando gli operatori sono diversi.
Esempio:
(1 = 1) * (4 > 2) + (3 < 9)
= (—1) * (—1) + (—1) = 0
L'espressione risulta falsa nonostante sia logicamente vera.
Per ovviare a questi problemi solitamente basta mettere un

AND	(1=1) * (4>3) = (—1) * (—1)	= 1	— vero
	(1=1) * (3>4) = (—1) * (0)	= 0	— falso
OR	(2=1)+(5>9) = (0)+(0)	= 0	— falso
	(2=1)+(5=5) = (0)+(—1)	= — 1	— vero
	(3>2)+(5=5) = (—1)+(—1)	= — 2	— vero
XOR	(3>2)—(5=5) = (—1)—(—1)	= 0	— falso
	(3>2)—(4>7) = (—1)—(0)	= — 1	— vero

Fig. 2

paio di parentesi al posto giusto. Nel caso dell'esempio:
(1 = 1) * ((4 > 2) + (3 < 9))
= (—1) * ((—1) + (—1))
= (—1) * (—2) = 2
L'espressione risulta ora vera come dovrebbe. In ogni caso consigliamo di assegnare dei valori alle variabili confrontate per assicurarsi del funzionamento delle espressioni logiche. Volutamente abbiamo trascurato l'operatore NOT, poiché per simularlo basta cambiare opportunamente i segni relazionali.

STEFANO DELLI PONTI

TI 99 MARKET

Finalmente SOFTWARE di altissimo livello ed assolutamente inedito per il Vostro TI 99/4A.

A sole L. 30.000 cadauno (10% di sconto agli ABBONATI) *sono disponibili i seguenti programmi:*

CON SUPPORTO DISCO DA 5"

RADAR
2000 COWBOY
SNOOPY
QUADRO X
SCOPA

LA DESCRIZIONE È A PAG. 25

CON SUPPORTO CASSETTA MAGNETICA:

TOMBOLA
+ GIROTONDO
ELESIUS
+ 1000 NOMI
TI 99 CALC

LA DESCRIZIONE È A PAG. 25

PER ORDINARE IN CONTRASSEGNO
(+ spese di spedizione):

TI 99 MARKET
Via Stefano Jacini, 4 - 20121 MILANO

IMPARIAMO A COSTRUIRE UN ARCADE GAME: GLI SFONDI NOBILI

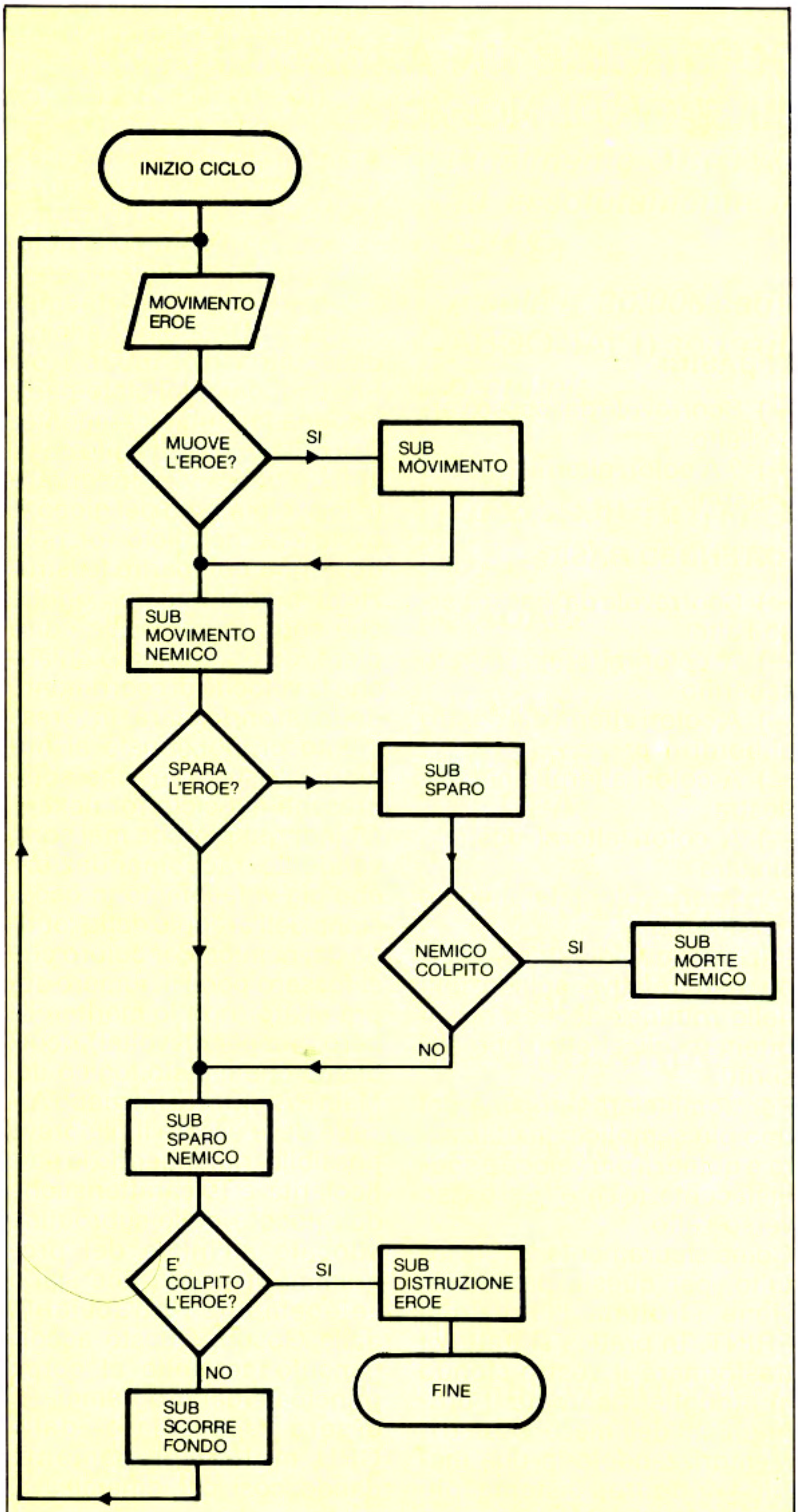
Rieccoci qui di nuovo ad affrontare la seconda parte del discorso sugli sfondi. Fermo restando che la soluzione degli sfondi fissi, è sempre la migliore in termini di velocità, parleremo questa volta di sfondi per così dire in movimento.
Il «per così dire», può sembrare un'espressione campata in aria ma vi assicuro che è stata da noi ben soppesata.
Infatti quando ci siamo posti il problema di fare un gioco dove lo sfondo fosse scorrevole, la prima cosa di cui ci siamo resi conto, era che il TI99, non possedendo l'accesso al linguaggio macchina, non poteva gestire lo Scroll (cioè il movimento dello schermo) in tre delle 4 direzioni.
A questo punto le possibilità erano due:
1° - o rinunciare, e questo non era nelle nostre intenzioni,
2° - oppure trovare altre strade, magari un poco macchinose ma che funzionassero.
A questo punto il discorso si spacca in due poichè per il TI-BASIC nasce un tipo di discorso, mentre per l'extended, oltre a poter sfruttare al meglio le possibilità offerte dal TI-BASIC, può offrire alcune possibilità in piu:
Per poter proseguire, vedremo subito di suddividere i tipi:

TI BASIC:

—) Scorrevole dal basso verso l'alto
—) ? A colori alterni a mezzo schermo

EXTENDED BASIC

—) Scorrevole dal basso verso l'alto
—) A colori alterni a mezzo schermo
—) A colori alterni dal centro ai bordi
—) A colori alterni sinistra - destra
—) A colori alterni destra - sinistra
—) Scorrevole lateralmente nei due sensi.
Iniziamo quindi la descrizione con relative spiegazioni delle routines necessarie ad ottenere gli effetti sopradescritti.
Per cominciare parliamo del tipo più semplice da utilizzare e quindi il più veloce e soddisfacente: lo scroll dal basso verso l'alto.
Come sicuramente avrete intuito, per questa procedura viene sfruttata l'istruzione PRINT. In pratica si tratta di trasformare il vostro sfondo in stringhe, ricordate il programma del mese scorso?. Memorizzandole in una matrice, oppure, sistema più lento in delle DATA, con un appropriato ciclo FOR NEST usando le print si fanno stampare le stringhe una per volta di seguito.
Prima di darvi il listato però, bisogna fare un chiarimento. Dato che lo scorrimento dello sfondo deve avvenire durante lo svolgimento del gioco, è ovvio che nel ciclo for next dovranno comparire le istruzioni macchina per quelle che sono le azioni da compiere, ed è altrettanto chiaro che la velocità di scorrimento dello sfondo sarà inversamente proporzionale al numero di istruzioni che sono presenti nel ciclo for next.
Quindi qui più che mai sono valide le raccomandazioni che già vi facemmo in occasione dell'articolo delle routine dei nemici. Per coloro che si fossero persi il numero sopra citato (non lo meriterebbero, ma per stavolta!) ricordiamo che il ciclo logico del Main Program di un gioco Arcade deve essere il più breve possibile, rimandando le particolarità e le caratteristiche del gioco a delle subroutine allocate all'inizio del programma in modo che siano velocemente rintracciabili dalle Gosub. Questo accorgimento permette al corpo principale del programma di essere «spazzolato» dalla CPU molto velocemente dando così più dinamica al gioco.
Ecco un FLOW-CHART particolareggiato del Main program di un SPACE-GAME:



Dopo avere dato uno sguardo all'indietro passiamo a questo fatidico Scroll:

```
100 call clear
110 CALL CHAR
    (33, "FFFF7E7E3C3C1818)
120 A$ =
"007E7E7E7E7E7E00"
130 FOR C=40 TO 88 STEP 8
140 CALL CHAR (C, A$)
150 NEXT C
160 FOR C=2 TO 9
170 CALL COLOR (C, C+I, I)
180 NEXT C
190 DATA 40, 3, 48, 4, 56, 5, 64,
6, 72, 7, 80, 8, 88, 9
200 FOR C=I TO 7
210 READ A (C, I), A (C, 2)
220 NEXT C
225 FOR S=I TO 10
230 FOR C=I TO 28
240 A$=A$&A (RND C,5+)
250 NEXT C
260 B$(S)=A$
270 A$=" "
280 NEXT S
290 CALL SCREEN (2)
300 CALL COLOR (1, 16, 2)
310 CALL HCHAR (1, 16, 33)
320 FOR C=I TO IO
330 PRINT B$(C)
340 CALL GCHAR(1, 16, X)
350 FOR S=I TO 7
360 IF X= A(S, I) THEN 390
370 NEXT S
380 GOTO 400
390 CALL COLOR (1, 16
A(S,2))
400 CALL HCHAR (1, 16, 33)
410 NEXT C
420 GOTO 320
```

Come avete potuto notare, il ciclo di movimento dello sfondo, in questo caso dalla linea 320 alla linea 420 è relativamente breve.
In questo esempio abbiamo inserito la costruzione dello sfondo con la funzione Randomize, tuttavia pensiamo che utilizzando i consigli del mese scorso, cioè costruendo uno sfondo composto da

30 o 40 righe, possiate rendere meglio l'idea di sfondo.
La routine di scorrimento in ogni caso è quella dell'esempio.
Le righe dalla 340 alla 390 servono per selezionare il colore di Backgraund della figura, in questo caso sono state inserite per rallentare lo sfondo.
Nel caso che, la figura che vi rappresenta, nell'esempio il triangolo, la vogliate piazzare in un posto diverso dalla 1ª linea, ricordate di inserire l'azzeramento del carattere occhio però di usare il sistema visto nell'articolo dei nemici per non cancellare lo sfondo.
Un esempio dell'applicazione di questa procedura, è stato usato dal programma Slalom pubblicata nel numero di settembre.
Come potete notare i vari procedimenti si incrociano vicendevolmente, è quindi importante conservare «religiosamente» tutti i numeri di TI99 NEWSOFT onde potere andare a consultare, qualora ve ne fosse la necessità, la routine desiderata.
Potremmo dire, che per questa puntata, l'argomento per quanto riguarda il Ti Basic, si esaurisce qui, comunque continuate a leggere, e questo per due motivi: primo: perché anche a chi non possiede l'extended in linea di massima il discorso può interessare, e poi perché, come promesso il mese scorso c'è in programma la modifica promessa di programmi ma questa la vediamo in seguito, parliamo adesso invece di quella che secondo me è la procedura più valida in extended, dopo naturalmente quella presentata prima.
In alternativa all'uso delle PRINT, possiamo costruire, degli sfondi scorrevoli bicolore. Un esempio di questa routine la troviamo nella cartuccia della SEGA «BUCK ROGERS».
Infatti nella prima parte del gioco si deve sorvolare un paesaggio a due colori. I colori, scambiandosi di posto uno con l'altro, accentuano l'idea del movimento, vediamo come.
Partiamo dal presupposto, che noi tracciamo 3 righe di caratteri queste 3 righe sono composte a loro volta da 2 metà.
Ora dopo avere fatto questa costruzione, usando un carattere diverso per ogni riga, poniamo la prima riga metà nera e metà bianca, mentre le altre due righe vanno completamente bianche.
Ecco quindi che ci siamo messi in posizione di inizio ciclo.
Il trucco dell'effetto di animazione è basato su di un effetto ottico, infatti per mezzo di un ciclo FOR-NEXT anneriamo la metà superiore delle righe, in sequenza, lasciando bianche le altre due, avremo che la riga nera è traslata, ciò ha dato l'effetto di movimento di tre caratteri. (Vedi fig. 1).
Generalizzando il concetto, se noi componiamo uno sfondo composto, per esempio, per metà azzurro tipo cielo, e l'altra metà con questo tipo di triplette, è possibile, dare l'effetto di una strada che scorre verso di voi, se in più volete l'effetto di curvatura basta che definiate un carattere tipo montagna, e lo mettiate in stringa e lo facciate scorrere al comando di curvare il veicolo.
Chiaramente, direte voi, ci va la routine di uno e dell'altra, e naturalmente, noi provvediamo ad inserire la routine nella cassetta.
A questo punto, direi che abbiamo trattato esaurientemente l'argomento sfondi, l'ultima parola spetta ai listati delle routine nella cassetta, vorrei inoltre qualche impressione dai nostri lettori, perciò scrivete, e soprattutto proponete argomenti da trattare, e le lettere indirizzatele alla attenzione della M.M.G. SOFTWARE.
Con questo vi salutiamo anticipando che la prossima volta parleremo delle routine dei punteggi, e salutando vi lascio il listato della modifica del CHAR-DEF.

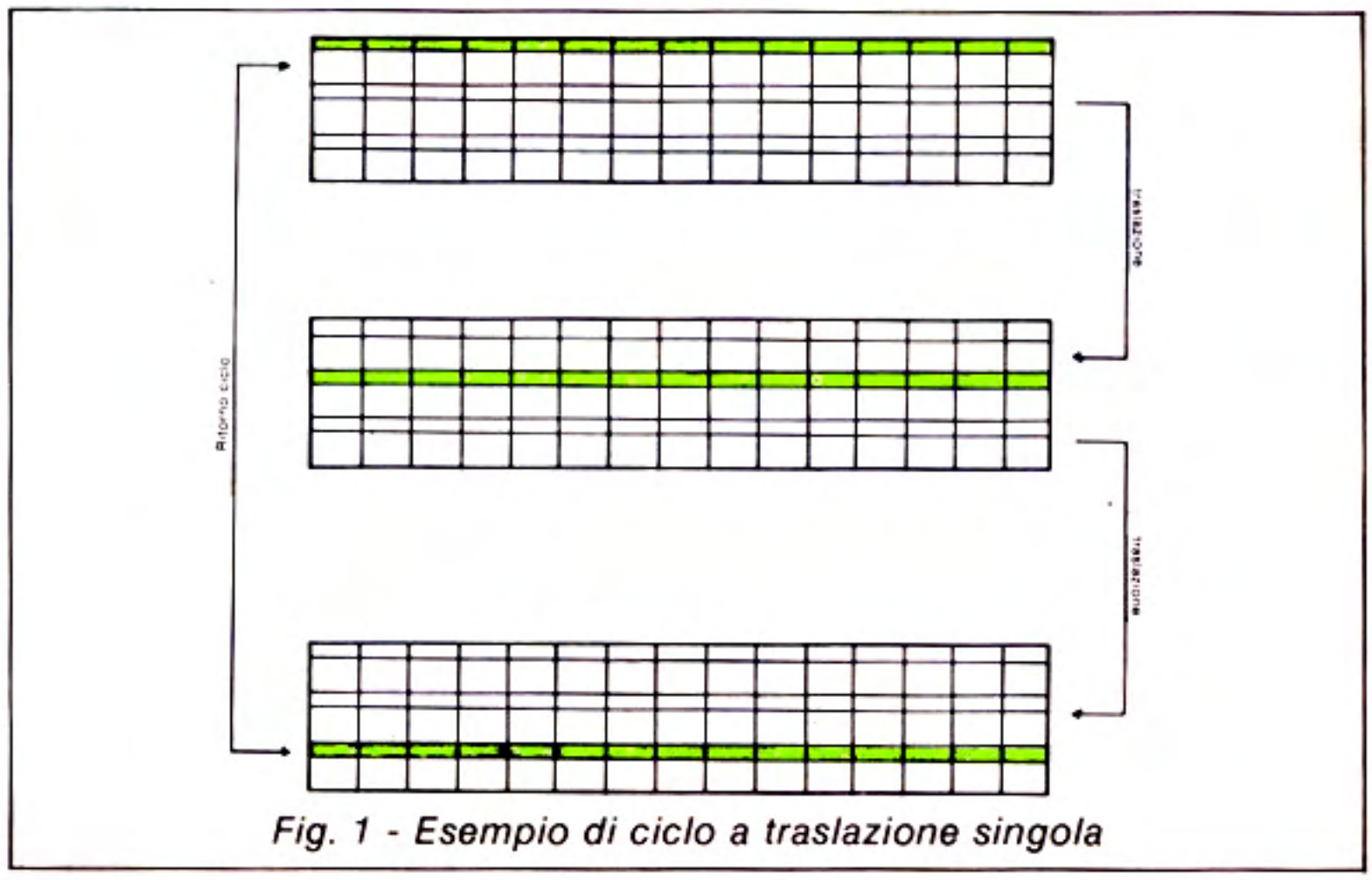


Fig. 1 - Esempio di ciclo a traslazione singola

QUASI UN'ARCADE: ROUTINE SFONDI IN MOVIMENTO

Come usare la routine inserita questo mese nella cassetta: la routine di scorrimento che inseriamo questo mese, tratta come si è detto dal gioco della SEGA «BUCK ROGERS», è stata compilata in EXTENDED per ovvi motivi di velocità, tenete presente che la routine originale è scritta in linguaggio macchina.
La routine, è già dotata di astronave in volo radente, tale astronave può spostarsi sia a destra che a sinistra premendo rispettivamente la D e la S. Questo per quanto ci competeva dato l'argomento dell'articolo, il resto dei movimenti, i nemici, e lo sparo con relative esplosioni, dovreste essere in grado di inserirle voi, anzi, direi che con poche aggiunte, la routine potrebbe trasformarsi in un ottimo gioco arcade.
Naturalmente la presente routine, con pochi cambiamenti, potrebbe trasformarsi da gioco spaziale in «POLE POSITION», il listato infatti contiene dei commenti, che vi permetteranno di andare a cambiare la forma agli SPRITE, ai caratteri che compongono le montagne che scorrono sull'orizzonte, che non sono composte da SPRITE.
In ogni caso non saremo noi a dirvi come usare la routine, che si presta ad innumerevoli utilizzi, quindi a voi scatenare la vostra fantasia, e poi i risultati mandateceli e se saranno validi penseremo noi a pubblicarveli.

MAURO MATTIAZZI

TI 99 MARKET è lieto di presentare una espansione di memoria da 32 K che non richiede il sistema di espansione Periferal Box.

Si collega semplicemente al computer ed è accessibile da EXTENDED BASIC, MINIMEMORY, LOGO, eccetera.

IL PREZZO E' DI 269.000 LIRE IVA COMPRESA.
PER GLI ABBONATI C'E' UNO SCONTO DEL 10%.

PER ORDINARLA IN CONTRASSEGNO:

TI 99 MARKET
Via Stefano Jacini, 4 - 20121 MILANO

UNA ROUTINE DI COLLOQUIO FRA "CHAR-DEF" E "TI TEXT FILE"

Ed ora eccoci alla tanto ventilata modifica del programma CHAR-DEF;
anche qui, come nei programmi gioco, le regole non cambiano, infatti, se ricordate, nel programma CHAR-DEF vi è un'opzione libera, la lettera «P» premendo la quale, si ottiene un messaggio di errore.
Allora, partendo appunto da questa porta lasciata libera per eventuali usi futuri, possiamo agganciare una subroutine che permetta al programma, all'occorrenza, di memorizzare su cassetta i codici esadecimali dei caratteri definiti.
Per fare questo, basta richiamare il programma CHAR-DEF, poi con il programma in memoria digitare le righe seguenti, e quindi, memorizzare il tutto su cassetta in modo di averlo sempre a disposizione all'occorrenza.

```
680 FOR CW=1 TO 4
685 PRINT#1:M1$(CW)
690 NEXT CW
695 CSR=CSR-4
700 IF CSR >= 0 THEN 705
    ELSE 730
705 CLOSE#1
710 CALL CLEAR
715 PRINT «HAI RAGGIUNTO
    IL NUMERO DEI CARAT-
    TERI
    CHE HAI DIGITATO
    ALL'INIZIO»
720 PRINT «PUOI PASSARE
    AL PROGRAMMA STRIN-
    GHE»
725 CALL CLEAR
728 END
```

```
 730 GOTO 465
1720 CALL CLEAR
1730 PRINT «DEVI COMPOR-
     RE
     SFONDI CON IL PRG.
     STRINGHE?»
1740 INPUT RISP$
1750 IF (RISP$ = «S») +
     (RISP$ = «s») THEN 1760
     ELSE 1820
1760 OPEN # 1: «CS1»,
     SEQUENTIAL, INTER-
     NAL,
     OUTPUT, FIXED 128
1770 CALL CLEAR
1780 INPUT «QUANTI CARAT-
     TERI
     DEVI DEFINIRE?»: CSR
1790 IF CSR > 120 THEN 1780
1800 BKL = 152 - CSR
1810 PRINT # 1: CSR, BKL
1820 CALL CLEAR
1830 RETURN
```

Questo è quanto, per il programma CHAR-DEF, di seguito, è necessario un adattamento del programma stringhe, in modo che possa leggere i caratteri che il CHAR-DEF memorizza su cassetta.
Per la modifica, seguire gli stessi accorgimenti usati per il CHAR-DEF:

```
175 GOSUB 660
177 IF FLIP = 1 THEN 350
660 PRINT «RICHIAMI I
    CARATTERI DA
    CASSETTA?»
670 INPUT CAS$
680 IF (CAS$ = «S») + (CAS$
    = «s»)
    THEN 685 ELSE 760
685 OPEN # 1: «CS1», INPUT,
    SEQUENTIAL, INTERNAL,
    FIXED 128
690 INPUT # 1: A, B
```

```
700 FOR C = B TO B+A - 1
710 INPUT #1 : A$
720 CALL CHAR (C, A$)
730 NEXT C
735 CLOSE # 1
740 FLIP = 1
750 GOTO 770
760 FLIP = 0
770 RETURN
```

Queste erano le semplici righe da aggiungere ai due programmi per renderli compatibili uno con l'altro.
Ed ora per chiudere, poche righe per l'uso della nuova opzione.
Dopo avere caricato il CHAR-DEF, se userete l'opzione, alla domanda che vi farà il computer in merito dovrete rispondere «S» oppure «s». A questo punto, ogni volta che avrete definito 4 caratteri diversi, premendo la P il programma provvederà a memorizzare i file che contengono i codici esadecimali.
Quando il numero dei caratteri definiti e memorizzati sarà uguale al numero da voi digitato all'inizio del programma, l'esecuzione del medesimo sarà sospesa dal computer.
A questo punto, basterà caricare il programma STRINGHE, e dopo avere risposto alle domande iniziali, potete comporre lo sfondo con i caratteri in precedenza definiti con il CHAR-DEE.
A questo punto ho proprio finito, e a questo punto... buona elaborazione a tutti e a rileggerci al prossimo mese.

MAURO MATTIAZZI

MAPPA RAM

Nel numero 1 (Settembre 84) abbiamo analizzato a grandi linee (ed elogiato) le capacità della MINI MEMORY.
Ora vogliamo approfondire il discorso con una serie di articoli che, di mese in mese, spiegheranno ed amplieranno quanto scritto sul troppo sintetico manualetto, fornito con il modulo. Naturalmente, se avete dei problemi specifici, comunicatecelo: noi cercheremo di accontentare tutti, rispondendovi tramite la rubrica.
In questo numero parliamo della CPU RAM, che si aggiunge alla memoria standard quando il modulo è inserito. Consigliamo di fare riferimento al suddetto manuale (e in particolare alla APPENDIX D) e premettiamo che considereremo caricato il LINE-BY-LINE ASSEMBLER.
Lo spazio che va dalla locazione >7000 alla >7116 è usato dal sistema operativo, dalle routines ROM e GROM, e dal programma NEW. Alcuni indirizzi sono però di grande utilità per il programmatore: vediamoli.
>7002->7011 contengono gli identificatori degli argomenti; ogni byte indica il tipo di parametro passato tramite "CALL LINK", secondo un codice spiegato a pag. 26 del manuale. I bytes sono 15 per i 15 argomenti permessi dall'istruzione.
I valori sono aggiornati automaticamente quando vengono usate le interface utilities (NUMASG, STRASG, NUMREF, etc.), altrimenti deve pensarci l'utilizzatore.
>701C, >701E contengono rispettivamente il primo (FFAM) e l'ultimo (LFAM) indirizzo libero nella Medium memory, cioè nei 4K che stiamo considerando.
Di particolare interesse si rivela la seconda delle due locazioni, in quanto la NAME LINK ROUTINE (che si occupa della ricerca di nome e indirizzo del programma assembly chiamato e che successivamente trasferisce il controllo ad esso) parte dall'indirizzo contenuto in questa per effettuare la ricerca; bisogna quindi ricordarsi di aggiornarlo (sottraendo 8 al valore trovato), con l'aggiunta di nuovi nomi alla REF-/DEF TABLE, perché altrimenti non vengono trovati, ed è segnalato l'errore "PROGRAM NOT FOUND".
>7020 contiene l'indirizzo di partenza dell'ultimo programma chiamato, cambiato di volta in volta dalla N.L.R.; se noi, quando ci viene chiesto il PROGRAM NAME, invece di digitarlo, schiacciamo immediatamente l'ENTER, verrà fatto partire il programma il cui entry point si trova in questa locazione.
>7022->7024; >7026->7028: in essi si trovano il primo e l'ultimo indirizzo libero della High memory e della Low memory. Ricordiamo che, quando l'espansione di memoria è connessa, c'è la possibilità di renderla «invisibile», in modo che il sistema non riconosca la sua presenza, semplicemente azzerando questi 4 bytes.
>70B8->70D6: è il WORKSPACE, l'area di lavoro dove si trovano i 16 registi (a meno che non venga cambiata locazione dal programmatore cn l'istruzione Assembler LWPI). Il registro 0 (RO) è situato nelle locazioni >70B8-9, R1 in >70BA-B e così via. E' molto utile sapere questo, poichè, in fase di debugging, diventa necessario conoscere i contenuti dei vari registri così com'è indispensabile conoscere i contenuti delle variabili programmando in BASIC.
Nelle locazioni da >7118 a >7CD4 è contenuto il programma LINE-BY-LINE ASSEMBLER (NEW e OLD).

Tabella 1 - Mappa RAM della mini Memory con Assembler caricato.

7000	RAM con puntatori, buffers, etc.
7118	Line-By-Line Assembler
7CD8	Symbol Table
7D00	User Program
7FE8	REF/DEF TABLE

Esso inizia in realtà all'indirizzo >71A6 ma alcune subroutines sono situate prima dell'entry point. Quando lavoriamo in Assembler è questo il programma che si occupa della traduzione, che controlla tutto ciò che facciamo, segnalandoci gli errori quando necessario. In uno dei prossimi numeri lo analizzeremo più dettagliatamente, spiegando le sue varie routines, ma per ora ci limiteremo a parlarvi di due locazioni.
>71D2 contiene il byte che, ripetuto 8 volte da una apposita routine, definisce il carattere del cursore lampeggiante. Non si potrà quindi ridefinirlo a piacere, ma si può comunque modificarlo, tenendo conto che essendo in TEXT MODE gli ultimi due bit di ogni pixriga non vengono considerati.
>722E contiene l'indirizzo (inizialmente >7D00) che viene immediatamente visualizzato quando viene fatto partire il programma NEW (o OLD) e dal quale solitamente si inizia la programmazione Assembler. Può essere cambiatofacendo attenzione alla nuova locazione.
Dall'indirizzo >7CD6 si trova il programma LINES (l'indirizzo di partenza è >7D9E) al quale ci si sovrappone inevitabilmente se si programma in Assembler.
Lo spazio da >7CD8 a >7CFE viene riservato alla SYMBOL TABLE, la zona di memoria dove sono memorizzati i nomi e gli indirizzi delle LABLE (di due lettere al massimo) che usiamo durante la programmazione. Ad ogni lable sono riservati 4 bytes (due per il nome + due per l'indirizzo) e perciò nominalmente ne sono permesse 10 (ma la S.T. può essere più lunga o più corta a seconda di dove è situato il primo byte del programma).
Da >7D00 abbiamo lo spazio per i nostri programmini, che devono essere proprio "ini" a causa dell'esiguo spazio disponibile (circa 760 bytes). Comunque è meglio che niente (dove, parlando di Assembler, niente = configurazione standard).
La REF/DEF TABLE inizia all'indirizzo segnato nella locazione >701E (LFAM) ed è inizialmente >7FE8.
Qui si trovano nome ed entry point del programma dimostrativo LINES (>7FE8), di OLD (>7FFO) e di NEW (>7FF8).
L'USER PROGRAM NAME può essere immagazzinato in >7FE8 poichè, in ogni caso, appena si modificano in qualche modo le locazioni disponibili, il LINES diventa inutilizzabile. Per altri nomi si deve aggiornare l'LFAM opportunamente e memorizzarli nei bytes immediatamente precedenti a quelli anzidetti.
Per questo numero abbiamo terminato, ma il discorso, certamente complesso, verrà completato nei prossimi articoli.
In particolare nel prossimo parleremo della ROM e delle sue routines principali (come la NAME LINK ROUTINE).

STEFANO DELLI PONTI

Tabella 2 - REF/DEF TABLE

7FE8	4C49	*LI*	nome
7FEA	4E45	*NE*	
7FEC	5320	*S*	
7FEE	7D9E		indirizzo
7FF0	4F4C	*OL*	nome
7FF2	4420	*D*	
7FF4	2020		
8FF6	71AC		indirizzo
7FF8	4E45	*NE*	nome
7FFA	5720	*W*	
7FFC	2020		
7FFE	71A6		indirizzo

LOADER CARATTERI E CARATTERI LINE

Avete mai pensato alla possibilità di definire tutti i caratteri, anzichè con delle enormi quantità di CALL CHAR o DATA con una sola istruzione che carica da nastro le pattern dei caratteri nello stesso modo con cui caricate i vostri programmi? Bene se siete in possesso della **Mini Memory** ora potete fare anche questo.

IL PROGRAMMA

Il programma, scritto in BASIC per Mini Memory, carica a partire dalla locazione 7D tre programmi assembler richiamabili con:
CALL LINK ("SAVE")
CALL LINK ("LOAD")
CALL LINK ("LARGE")
"SAVE"
Con il programma "SAVE" potete salvare su nastro le pattern dei caratteri con codice da 32 a 127, in seguito vi spiegheremo come salvare anche i caratteri da 128 a 159.

"LOAD"

Ora che sapete come salvare i caratteri possiamo passare alla seconda parte ovvero come caricarli in memoria;
Per caricare i caratteri che avete salvato su nastro dovete battere CALL LINK ("LOAD") e seguire le istruzioni che compaiono sul video.
Ora che sapete tutto facciamo un esempio:
Avete visto i nuovi caratteri LINE pubblicati in questo numero di TI 99 NEWSOFT e vorrreste implementarli al programma che avete appena terminato. Se avete pazienza potete aspettare il numero di NEWSOFT in cui pubblicheremo il programma di emergenza per registratore, se invece avete fretta di vedere come sta il vostro programma con i nostri caratteri seguite queste istruzioni.
Caricate il programma LOADER CARATTERI e date il RUN. Quando avete finito caricate il programma di "caratteri LINE". Fatto? Ora modificate la linea 90 con
90 CALL LINK ("SAVE")
Date il run e seguite le istruzioni che compaiono sul video per salvare i caratteri.
Caricate a questo punto il vostro programma e inserite come prima linea
1 CALL LINK ("LOAD")
Ora date nuovamente RUN e seguite le istruzioni per il registratore caricando i dati che avete appena salvato. A questo punto, se avete seguito attentamente le istruzioni il vostro programma girerà con caratteri LINE.

Come ho già detto il programma lavora sui caratteri da 32 a 127 questo perché la zona di memoria dei caratteri da 128 a 159, se questi non sono ancora stati definiti con una CALL CHAR, viene usata dall'interprete BASIC come stack. Se volete salvare anche questi caratteri dovete eseguire la seguente modifica al programma "LOADER CARATTERI"
la linea 400
400 DATA
040000003006003...
diventa
400 DATA
040000004006003...
Per la "SAVE" non ci sono variazioni, mentre per il "LOAD"
bisogna scrivere:
1 CALL CHAR (159,"")
2 CALL LINK ("LOAD")
la call char serve spostare i puntatori di memoria dello stack.

"LARGE"

Avete mai fatto caso che i caratteri del titolo principalmente sono leggermente più grandi di quelli del BASIC?
Il TI 99 ha infatti 2 set di caratteri maiuscoli e con il programma "LARGE" potete richiamare il set di caratteri del titolo di testa.
Riprendete il vostro programma; ora cambiate la linea con
1 CALL LINK ("LARGE")
Quando farete girare il programma questo avrà i caratteri del titolo di testa.

RICCARDO ROSSI

LOADER CARATTERI LISTATO ASSEMBLER

```
PAB     DATA>0400       Inizio area di memoria da
        DATA>0000
        DATA>0300       Numero di Bytes da salvare
        DATA>6003
TEX     TEXT 'CS1'      Nome dispositivo
LOAD    LI R1,>500      Call Link ("LOAD")
        JMP A           Carica in memoria i caratteri
SAVE    LI R1,>600      Call Link ("SAVE")
A       LI R0,>302      Salva i caratteri in memoria
                        Indirizzo PAB in VDP
        BLWP ∂ VSBW
        INCT R0
        Li, R1, PAB
        LI R2,12
        BLWP ∂ VMBW
        LI R0,>834A
        LI R1, TEX      Scrive in FAC il nome
        MOV ⌗ R1+, ⌗R0* del dispositivo
        MOV ⌗R1, ⌗R0
        LI R1,3
        MOV R1, ∂ > 8354
        LI R1,>30F
        MOV R1, ∂ >8356
        CLR ∂>83D0
        MOVB ª >83D0, ∂ 837C
        LI R1, 800
        MOVB R1, ∂ GPLLNK  Chiama la routine di
        DATA > 3D          Attivazione CS1
        B ⌗R11             Ritorna in Basic
LARGE   LI R0,>400         Call Link ("LARGE")
        MOV R0, ∂ >834A    Carica i caratteri del
        BLWP > 16
        B ⌗R11             Ritorna in Basic
```

CARATTERI "LINE" CON L'EXTENDED

Nell'articolo precedente abbiamo detto cosa può fare chi ha la Mini Memory per definire i caratteri all'esterno di un programma. Con l'extended Basic potete farlo senza ricorrere all'assembler:
caricate il programma di "caratteri line" alla linea 90 scrivete:
90 RUN "CS1"
Dopo che avrete dato il RUN, prima verranno ridefiniti i caratteri, poi appariranno i messaggi per il caricamento del programma. Il programma che caricherete manterrà i caratteri così ridefiniti fino a che non verrà fermato.

RICCARDO ROSSI

Avete acquistato il Numero di Novembre

Ma sapete che sono usciti i Numeri di Giugno-Settembre e Ottobre?

Per riceverli direttamente a casa vostra, scriveteci! Ve li invieremo in contrassegno a L. 10000 (+spese cadauno).

TI 99 NEWSOFT
Via S. Jacini, 4
20121 MILANO

MUOVERE GLI SPRITES CON I JOYSTICK

Questo mese affronteremo il problema: come muovere uno sprites, tramite una CALL MOTION, con i joistick, discutendo vari metodi.
Uno fra i più usati è quello di creare una tabella di confronto di valori affiancata da una tabella di velocità.
Per tabella di confronto intendiamo una serie di operazioni logiche eseguite dall'istruzione basic IF THEN ELSE.
Le variabili di ritorno del joystick sono due, e danno rispettivamente valori di X e di Y, a seconda che la leva sia spostata da una parte anziche da un'altra.
I valori di ritorno possono essere tre tipi:
— +4 il joystick è stato posizionato in un campo positivo.
— —4 il joystick è stato posizionato in un campo negativo.
— O il joystick è in posizione di riposo.
Le combinazioni di tali valori tra X e Y danno in totale 8 possibili direzioni, e ogni direzione da un valore ben preciso.
Una tabella delle 8 possibili combinazioni con i relativi valori potrete trovarla sul manuale al paragrafo CALL JOYST.
Detto questo risulta semplice strutturare una piccola routine che ci permetta di convertire tali valori, in velocità di riga e colonna per il movimento dello sprite.
Se una velocità di riga o colonna di soli 4 pixel ci sembra troppo lenta, possiamo moltiplicarla per 4,5 o un numero più alto, facendo attenzione a non superare il limite positivo di 127 e quello negativo — 128.

I PROGRAMMI

Come esempio pratico vi presentiamo due piccoli programmi che potranno, se lo desiderate, sostituire i vecchi metodi da voi usati, sempre che voi non abbiate trovato un modo migliore!
Il primo listato, figura 1, ci illustra il principio generale; quello di inserire direttamente il valore di ritorno del joystick nella CALL MOTION, avendo l'accortezza di invertire le X con le Y e mettendo il segno — davanti a Y, perché il valore di riga che il joystick fornisce, corrisponde al valore positivo di colonna della CALL MOTION, per la stessa ragione dovremo cambiare segno alle coordinate di colonna Y.
Nel secondo listato, contenuto in figura 2, possiamo vedere una semplice ma ingegnosa modifica alla regola generale sopra citata.
Il programma genera uno sprite con la forma di un asterisco e muovendo la leva del joystick possiamo ottenere un simpatico effetto di accellerazione e decellerazione.
Quando lo sprite raggiunge la massima velocità di riga, colonna o entrambe, lo sprite si ferma ripartendo dalla velocità di Y, X = 0.
Questo effetto è stato ottenuto senza l'ausilio di nessun ciclo logico IF THEN ELSE, a tutto guadagno della velocità di esecuzione.
L'espressione di comparazione è ottenuta sfruttando il principio:
se la condizione è verificata il valore di ritorno è 0, se la condizione non è verificata è —1.

EZIO MONTINI

```
100 CALL CLEAR :: CALL SCREEN(13)
:: CALL MAGNIFY(2) :: CALL SPRITE
(#1,42,16,100,100)

110 CALL JOYST(1,X,Y) :: CALL MOT
ION(#1,-Y*4,X*4) :: GOTO 110
```

```
100 CALL CLEAR :: CALL SCREEN(13)
:: CALL MAGNIFY(2) :: CALL SPRITE
(#1,42,16,100,100)

110 CALL JOYST(1,C,R) :: X=(X+C)*
-(ABS(X)<124) :: Y=(Y-R)*-(ABS(Y)
<124) :: CALL MOTION(#1,Y,X) :: G
OTO 110
```

DUE O TRE COSE CHE SAPPIAMO DI LUI

Presenteremo in questa rubrica piccole cose che, speriamo, ci aiuteranno a comprendere meglio il nostro calcolatore e possono facilitarci la programmazione.

L'ISTRUZIONE INPUT.

Questa istruzione, presente del resto in tutti i dialetti Basic, offre la possibilità di ottenere dati dalla tastiera ma non è immune da imperfezioni e può creare degli equivoci nel caso di conflitti tra dati letterali e numerici.
In Extended Basic abbiamo l'istruzione Accept At, molto più potente e in grado i di definire l'immisione di dati numerici, di stringa oltre a delimitare il campo di accettazione e la posizione sullo schermo.
Forse non tutti tra noi si sono accorti che l'istruzione Call Key serve egregiamente a simulare l'input di dati e può, per certi versi,sostituire degnamente l'Accept At dell'Estended.

INPUT NUMERICO

Una delle cose essenziali in un programma è stabilire se ad un certo input sia correlata una risposta numerica o letterale: a chi di voi non è mai capitato il fatidico: Input "Grado di difficoltà' ?": G.
Al dannato G il timido Abele risponde — scarso, mentre l'orgoglioso Anselmo digita — ATROCE.
Oppure, peggio, capita di rispondere 1-20 Enter, o anche 1999782.786.
E veder cosi il bieco punto interrogativo che occhieggia celando chissà mai quali nefandezze con l'aria di dire: niente so! e se dir niente mi danneggia, non so neppure quello.
E il suo empio compare — WARNING: INPUT ERROR, TRY AGAIN sempre pronto a spalleggiarlo con aria irridente e lo zufolo annichilatore — poi chissa perché WARNING ci fa sempre pensare ai film di John Wayne e ai depositi di nitroglicerina, è sempre un avvertimento scabroso, ci lascia sulle spine —.
Poi sono anatemi biblici, le lettere di fuoco e sacchi di cenere sul programmatore che si è recato in rifugio atomico in Svizzera.
Basta giuriamo che aboliremo l'istruzione Input ed useremo, fin che tempo ce lo permetta, la Call Key.

```
100 CALL CLEAR
110 GOTO 1000
120 B$=""
125 D=0
130 C=LEN(A$)+3
131 IF C>30 THEN 135
132 GOTO 140
135 A$=A$&"."
140 PRINT A$;
150 GOTO 170
160 CALL SOUND(100,1000,10)
170 CALL KEY(0,A,S)
180 IF S=0 THEN 170
190 IF A=13 THEN 250
200 IF (A<48)+(A>57)THEN 160
210 B$=B$&CHR$(A)
215 IF LEN(B$)>2 THEN 160
217 IF C>28 THEN 219
218 GOTO 220
219 C=C-28
220 CALL HCHAR(24,C+LEN(B$)-1,A)
230 GOTO 170
240 B=0
250 B=VAL(B$)
255 D=D+B
260 IF (D>20)+(D<1)THEN 100
270 RETURN
1000 A$="GRADO DI DIFFICOLTA'? (1-20)"
1010 GOSUB 120
```

CALL KEY

L'istruzione Call Key è quella che più si presta all'accettazione di comandi dalla tastiera; è interessante notare che non vengono soltanto riconosciuti da essa i codici normali ASCII ma anche i vari: SCHIFT A (aid), G (ins), T (erase), D (freccia a destra), E (freccia in su), ENTER, Z (indietro), F (delete), R (redo), S (freccia a sinistra), X (freccia in giù), V (cmd), W (begin).

Con i rispettivi codici numerici:
1, 4, 7, 9, 11, 13, 15, 3, 6, 8, 10, 12, 14.

Detto questo, argomento forse non pienamente spiegato nel manuale del nostro calcolatore, passiamo all'utilizzo introducendo una routine, adattata alle nostre esigenze, che sostituisca l'input numerico.

Notiamo che alla riga 190 viene effettuata un verifica per controllare che il codice di ritorno sia 13, nel qual caso il calcolatore accetta Enter come ingresso, e, dopo una verifica sulla dimensione del dato, ritorna al programma principale.
Basta per ora sulla Call Key, ma avremo modo di ritornare su questa elegante istruzione.

PAOLO CIVARDI

GUARDIA E LADRI

Voi siete dei ladri, nel gioco si intende, e dovete correre per la città rubando più sacchi di dollari che potete e cercando di sfuggire al forte braccio della giustizia che vi insegue cercando di cogliervi con le mani nel sacco.
La legge in questo gioco non prevede la perquisizione domiciliare, cosicchè una volta entrati nel vostro rifugio, non potrete essere più arrestati. Sfuggire alla giustizia diviene però, di volta in volta, sempre più difficile.
Per vostra fortuna la legge è molto generosa prevede infatti che per il furto di sacchi da 1000 $ la libertà condizionata possa essere concessa per tre volte.

Il meccanismo del gioco è molto semplice:
il ladro si controlla con il joystick (potete usare anche la tastiera modificando il programma come illustrato nel riquadro) la guardia si trova a destra sullo schermo e rimane immobile fino a che il ladro non ruba i primi 1000 dollari. A questo punto inizia uno spietato inseguimento ostacolato da muri che né il ladro né la guardia possono scavalcare. L'unico modo per sfuggire alla guardia e mettere in salvo la refurtiva, è quello di scappare cercando di raggiungere il vostro rifugio che si trova in basso a destra. Se riuscirete a non farvi prendere guadagnerete la possibilità di compiere un altro furto.
Il programma gira in TI Basic — Volendo, gira anche in Extended ma solo con l'espansione da 32 K.
N.B. ogni riferimento a persone o fatti è puramente casuale.

RICCARDO ROSSI

GUARDIA E LADRI VARIABILI PRINCIPALI

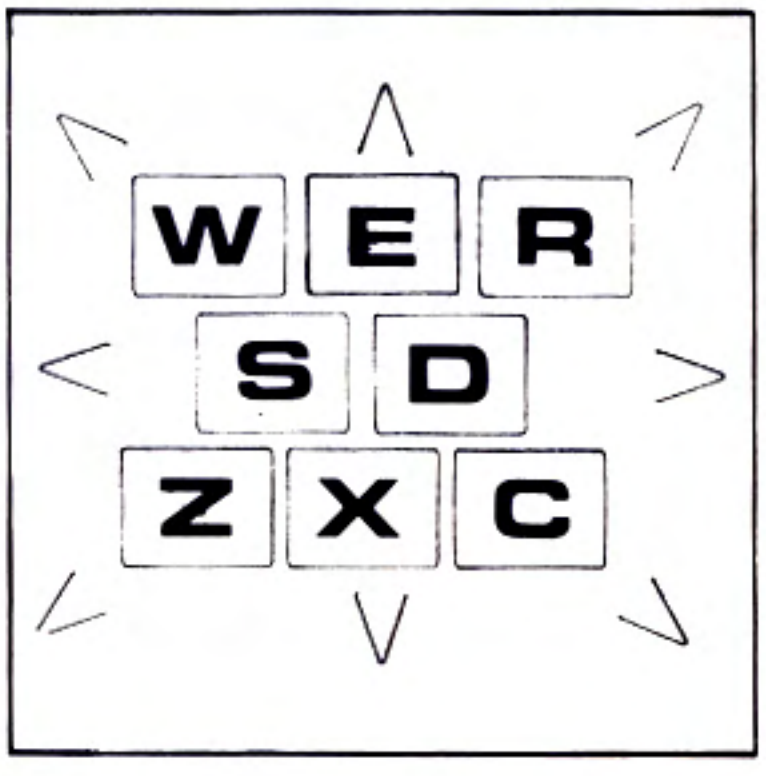
CITTA (33, 25)	pianta della città
AR$ (4)	arresti
CH, CH$	variabili per la definizione dei caratteri
MM	origine dei numeri casuali
AR	numero di arresti
COF	coefficiente di ritardo musicale
M, I, J	contatori
R, C	coordinate del ladro
GR, GC	coordinate della guardia
S	furto parziale
SS	furto totale
PM	furto del secolo
X, Y	lettura joystick
NR, NC	spostamento ladro
RN, CN	spostamento guardia
MUR	difficoltà

GUARDIA E LADRI ANALISI DEL LISTATO

10- 350	Ridefinizione caratteri
420- 480	Definizione colori
510- 650	Titolo principale
720- 790	Presentazione
800-1280	Disegno città
1300-1310	Controllo inizio del gioco
1320-1550	Controllo movimento del ladro
1560-1780	Controllo movimento della guardia
1790-1850	Cattura del ladro
1860-1960	Furto riuscito
1970-2000	Ritardo
2010-2130	Quadro finale
2140-2230	Musica del gioco
2240-2350	Musica della cattura
2360-2350	Musica del furto riuscito

MODIFICA PER LA TASTIERA

```
1300 CALL KEY (3, X, Y)
1310 IF Y = 0 THEN 1300
1320 CALL KEY (3, X, Y)
1325 Y = -4 * ((X=87) + (X=69) + (X=82)) + 4 * ((X=90) +
(X=88) + (X=67))
1326 X = -4 * ((X=82) + (X=68) + (X=67)) + 4 * ((X=87) +
(X=83) + (X=90))
```



LANDER

Questo programma è un'ottima versione per TI99 di un gioco da bar molto gettonato un paio di anni fa.
Richiede l'uso dell'**EXTENDED BASIC**, scelto per la possibilità degli SPRITES ed ha la capacità di pilotare il **Sintetizzatore Vocale.**

IL GIOCO

Dopo il RUN e la presentazione grafica inizia il gioco vero e proprio: Vi trovate nello spazio al comando di un piccolo modulo lunare e dovete atterrare dolcemente in un punto del suolo sottostante, determinato casualmente. Ciò non è tuttavia facile: il vostro modulo ha come nemico principale la forza di gravità, che vi costringe alla massima attenzione.
Nella parte alta dello schermo vi vengono mostrate alcune informazioni relative alla vostra navicella: velocità verticale (**VV**) velocità orizzontale (**VO**). Distanza dal punto di atterraggio (**X**), carburante (**F**) e numero di atterraggi effettuati (**A**).
Bisogna inoltre tener conto di quanto segue: se la velocità verticale è negativa significa che state dirigendovi verso l'alto, altrimenti state... cadendo! Quando la X è uguale a zero vuol dire che siete nel punto giusto per atterrare; quando è negativa dovete dirigervi alla vostra destra, se è positiva a sinistra.
Per pilotare il modulo si usano i tasti S, D, K, e L. S e D lo ruotano rispettivamente in senso antiorario e orario.
K e L mettono in azione i razzi di frenata e manovra (L da maggiore spinta). Logicamente l'azione dei razzi dipende dall'inclinazione della navicella.
Ogni rotazione del modulo comporta un consumo di 10 unità di carburante, come i razzi di minor intensità; gli altri consumano 20 unità: ma non c'è da preoccuparsi troppo, perché vi verrà fornito del "FUEL" ad ogni atterraggio riuscito. Terminata la riserva di carburante, cadrete inesorabilmente al suolo.
L'atterraggio deve essere effettuato nel punto X=0, con velocità verticale minore o uguale a 2 e con il modulo in posizione normale, altrimenti vi schianterete al suolo.
Se esplodete in un punto non visibile significa che la caduta ha creato un cratere nel suolo lunare (!).
Anche un tentativo di oltrepassare il bordo alto dello schermo causa la distruzione del vostro modulo.
Un'ultima avvertenza: dopo il quinto atterraggio la forza di gravità raddioppierà e addirittura triplicherà dopo il quindicesimo. Buona fortuna!

STEFANO
DELLI PONTI

VARIABILI PRINCIPALI

A	velocità verticale del modulo
AA(,)	incremento velocità verticale per i razzi
B	velocità orizzontale del modulo
BB(,)	incremento velocità orizzontale per i razzi
C	codice carattere del modulo
E	incremento velocità verticale=gravità
KK	distanza dal punto di atterraggio
LLL	punto di atterraggio determinato casualmente
UU	atterraggi effettuati
Z	carburante
WW	incremento carburante dopo ogni atterraggio

CARATTERISTICHE DEL PROGRAMMA

170-240	definizione caratteri: astronave, titolo, esplosione
250-370	presentazione
380-480	preparazione schermo gioco
490-540	inizializzazione variabili
570-780	loop principale
600	CALL KEY
900-930	esplosione, fine gioco

SPACE INVADERS

Non c'è molto da dire su quello che può essere definito il più famoso tra i giochi di tipo arcade.

Questa versione molto veloce e impegnativa, sfrutta ottimamente le capacità operative del TI 99/4A sulle operazioni di stringa e sugli sprites.

IL GIOCO

Il programma richiede il modulo EXTENDED BASIC e i Joysticks (il tasto ALPHA LOCK deve pertanto essere disinserito): è possibile però abilitare la tastiera con le modifiche illustrate più avanti.
Dopo la videata di presentazione occorre scegliere uno fra i tre livelli di gioco.
Il giocatore deve eliminare gli invasori cercando di evitare i missili, che gli vengono lanciati sempre più da vicino, spostandosi con la leva del telecomando e premendo il pulsante di sparo.

Ogni tanto, preannunciato da un suono di campanello comparirà, sopra la schiera di invasori, un ufo nemico che dovrà essere abbattuto con un tiro.
Al termine del gioco, dopo l'immancabile invasione verrà presentato il punteggio raggiunto peraltro visibile in alto sullo schermo durante il gioco: a questo punto, per continuare a giocare, occorre premere il pulsante di sparo, per finire un qualsiasi tasto.
Per abilitare la tastiera si dovrà eliminare il REMARCK alla linea 3000 e scrivere 13 anziché 16 alla linea 500; si potranno, utilizzare i tasti S e D per lo spostamento dell'astronave e il tasto Q per lo sparo.

ALBERTO STRAFILE

VARIABILI UTILIZZATE

Variabile	Funzione
INV$(4)	Tabella con i 4 tipi di invasori
R$(4)	Stringhe contenenti le righe di invasori
DIR(16,2)	Matrice con le direzioni per le esplosioni
DIF$, DIF	Livello di difficoltà
PUNTI	Accumulatore del punteggio
AST	N° astronavi ancora in gioco
C,R	Coordinate della posizione del blocco di invasori
A,V,S,I, DELAY	Comodi per i cicli FOR NEXT
K, S	Comodi per le CALL KEY
X	Variabile per il movimento dell'astronave
COL	Posizione dell'astronave
CO	Indicatore di coincidenza

ANALISI DEL LISTATO

Righe	Funzione
10- 160	Definizione matrici e presentazione titolo
170- 410	Definizione caratteri grafici
420- 480	Richiesta del livello di difficoltà
490- 520	Partenza astronave
530- 580	Costruzione righe invasori
590- 720	Loop principale: spostamento righe invasori, controllo joysticks e pulsante di sparo, schedulazione sparo invasori e partenza ufo
730- 830	Tiro astronave e controllo invasore colpito
840- 960	Tiro invasori e controllo astronave colpita
970-1070	Ricostruzione di una riga con eliminazione di un invasore colpito
1080-1200	Astronave colpita
1210-1290	Controllo invasione ed eventuale fine gioco
1300-1390	Partenza ufo
1400-1460	Tiro su ufo
1470-1570	Scoppio ufo
1580-1610	Suono campanello
1620-1710	Partenza astronavi
1720-1780	Fine gioco
1790-2000	Presentazione
3000	Abilitazione tastiera

STATISTICA E... TOTOCALCIO

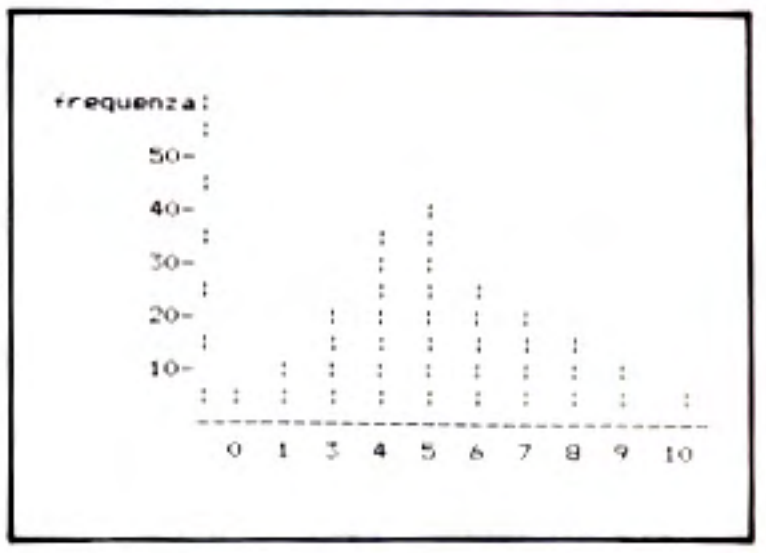
In un numero precedente avevamo introdotto la statistica come utilizzo del calcolatore, avevamo parlato di media e di deviazione standard.
In questa puntata pubblichiamo un programma per il calcolo di questi parametri insieme ai loro limiti di confidenza.
Perché i limiti di confidenza? Perché, tutto sommato dobbiamo ricordarci che, nella statistica, qualsiasi misura rappresenta sempre soltanto la stima di un valore corretto.
Nelle applicazioni statistiche la popolazione — ovvero la quantità di dati totali — è di solito molto grande mentre, abitualmente, è possibile, o semplicemente più conveniente disporre di una serie limitata di dati.
Pertanto, è poco prudente sbilanciarsi proclamando verità inesistenti, più umilmente di solito, si preferisce accettare un'ipotesi oppure rifiutarla consci di avere una certa possibilità percentuale di essere dalla parte della ragione o del torto.
Non è un modo di pensare subdolo e manicheo - è un tentativo per non prendere cantonate funeste.

A COSA PUO' SERVIRE LA STATISTICA

È evidente che la statistica, come tutte le scienze, non ha bisogno di una dimostrazione circa la sua utilità, sarebbe possibile solo fare discorsi filosofici su questo punto e ovviamente non è questa la sede adatta per introdurre disquisizioni metafisiche. Cercheremo pertanto di enunciare qualche concetto di questa scienza accompagnandolo con programmi di ultilità generale e tentando applicazioni immediate — scommettiamo che per applicazione immediate molti hanno pensato al totocalcio? — bene anche la schedina può essere una utile palestra per impiegare certi concetti, magari soltanto per dimostrare che in questo campo è solo necessaria la fortuna e la ragione non vale.

PROBABILITA' E DISTRIBUZIONE

Possiamo definire "probabilità" la possibilità che un dato evento avvenga; se per esempio gettiamo in aria una moneta due eventi sono possibili: che cada croce oppure testa. Definiamo pertanto la probabilità che esca croce come uguale a 1/2.
Ricorriamo ad un altro esempio: quali sono le probabilità che per un certo risultato in schedina esca l'1, la X o il 2? chiaramente ognuna di queste probabilità è uguale a 1/3. (La somma delle probabilità da sempre 1 come risultato).
Bene, ora che abbiamo parlato della teoria vediamo come vanno le cose in pratica. Parliamo di distribuzione e delle sue leggi.
Se gettiamo per dieci volte in aria la famosa monetina sarà certamente estremamente difficile che escano esattamente 5 teste e 5 croci.
Se però ripeteremo per un certo numero di volte questa procedura vedremo che la frequenza di uscita delle teste assumerà una distribuzione tipica (per inciso chiamata distribuzione normale o Gaussiana); il valore mediano sarà uguale a 5 (cioè uguale alla probabilità di 1/2 di cui si era parlato.



Esiste in statistica un metodo, detto di chi quadro, che permette di stabilire se esistono discordanze troppo forti dai valori attesi — nel qual caso è legittimo cominciare a nutrire seri dubbi sulla moneta e sul lanciatore — mentre nel caso di dati troppo coincidenti con quelli attesi il metodo ci avvertirà che qualcosa non funziona e i dati possono essere, come dire?, un po' addomesticati.

IL TEST CHI QUADRO

Alleghiamo un programma che ci permetterà di formulare ipotesi basate sulla probabilità e sulla attendibilità di alcune nostre supposizioni.
Alcune note al programma.
Ad un certo punto del programma vengono chiesti i gradi di libertà. Essi sono definiti come segue: v = numero di classi meno il numero di restrizioni.
Intendendo per numero di classi il numero di dati in esame, nel caso di una distribuzione normale come nel lancio di un dado, di una moneta, eccetera. Il numero di restrizioni sarà uguale a 1 nel caso che coincidano i totali delle frequenze attese con quelle osservate, a 2 quando coincidano anche le medie, e uguale a 3 qualora siano uguali anche le rispettive deviazioni standard.
Per tutto il resto crediamo che le istruzioni all'interno del programma siano abbastanza chiare.

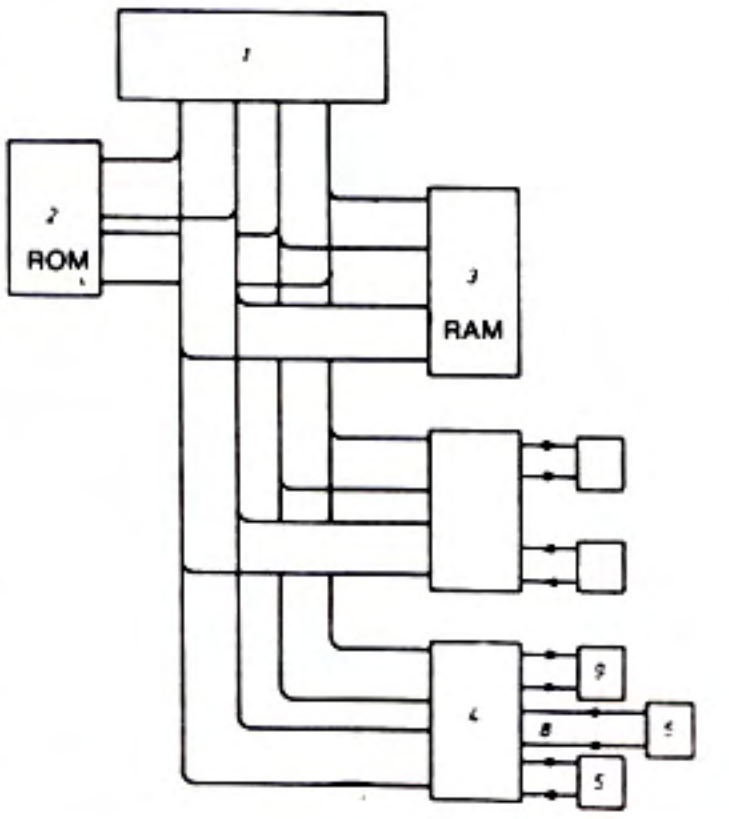
PAOLO CIVARDI

L'INVENZIONE DEL MESE

Dal presente numero di TI99 NEWSOFT iniziamo la presentazione dell'attività inventiva riguardante il settore dei computers, dei videogames ecc..., illustrando mensilmente un'invenzione felicemente realizzata e di risonanza internazionale.

"APPARECCHIATURA DI ADDESTRAMENTO AL MICROCOMPUTER" (Brevetto Inglese 040535)

La presente invenzione si riferisce ad un'apparecchiatura di insegnamento e/o di addestramento al microcomputer ed ha lo scopo di dimostrare quali sono le funzioni fondamentali di un microcomputer in modo semplificato.
L'apparecchiatura comprende in combinazione i seguenti elementi: un microprocessore (1); una memoria di sola lettura (2) contenente i programmi del monitor; una memoria dei programmi ed una memoria dei dati comprendenti una o più memorie ad accesso casuale (3); un dispositivo di interfaccia ingresso/uscita (4); mezzi di commutazione manuale d'ingresso (5), (6), e mezzi (9), tali come lampade o diodi che emettono luce, per visualizzare l'uscita dei dati.
L'apparecchiatura è caratterizzata dal fatto di avere un solo modo di indirizzamento, una lunghezza fissa di parola di 8 bits per le istruzioni del programma e per i dati che devono essere elaborati, ed è capace di trattare un numero di istruzioni non inferiore ad 8 ma non superiore a 20. I programmi del monitor consentono l'entrata dei dati e del programma mediante mezzi di commutazione dell'ingresso e permette che i programmi così entrati vengano provati ed elaborati. Possono essere previsti dei mezzi per indicare un errore dell'operatore, il che da luogo ad entrate non corrette.



COYOTE

Munitevi di 2 Joysticks e andiamo a giocare... ad acchiapparello. Un giocatore manovra il coyote mentre l'altro dirige la pecora. Scegliete quante volte il coyote deve assalire la pecora (numero di turni) e via! Il campo si delinea. Il coyote muoverà per afferrare la pecora, la pecora per sfuggire al coyote. Un cane da pastore si aggira per il campo pronto ad impartire una sonora lezione al coyote, assegnando contemporaneamente punti alla pecora. La pecora può sfuggire e con un po' di abilità può anche nascondersi fuori campo.
In tal caso il coyote non potrà prenderla, ma potrà a sua volta andare a caccia del cane per ridurre il punteggio della pecora. Terminato il numero dei turni stabilito, i giocatori si scambiano i ruoli.
Vince infine chi manovrando la pecora, avrà totalizzato un punteggio maggiore, resistendo il più a lungo possibile agli assalti del coyote. Il punteggio realizzato in ogni turno appare in basso a destra quando la pecora viene catturata. Il numero di turni giocati appare in alto a destra dello schermo.

NOTE TECNICHE

Ottenere la velocità in un gioco realizzato in Extended Basic non è cosa semplice non appena gli sprite in azione sono più di due. Il miglior risultato, anche se non ottimale dal punto di vista della precisione, si ottiene adoperando l'istruzione CALL COINC (ALL, C) invece che tre istruzioni separate per la coincidenza tra cane e pecora, cane e coyote, pecora e coyote.
La routine principale deve inoltre contenere il minor numero di istruzioni possibile intercalate tra due CALL JOYST consecutivi, e tra due CALL COINC consecutivi.
Il problema si sposta poi a capire quali due sprite tra i tre sono in coincidenza. Se gli sprite non uscissero da una parte dello schermo per ricomparire dal lato opposto, basterebbe vedere la distanza tra le coppie di sprite. Nel nostro caso, data la grande mobilità degli sprite che attraversano lo schermo e data la limitazione a 32767 del quadrato della distanza tra due sprite, può accadere che si verifichi coincidenza quando uno degli sprite appare a metà su un bordo dello schermo.
In tal caso, esso può figurare posizionato dall'altro lato, mentre l'altro sprite è ancora al di qua del limite dello schermo. Ciò porta ad una alterazione notevole della distanza calcolata. Le linee di programma da 310 a 350 offrono una soluzione, anche se non ideale, a questo difficile problema.
Ai lettori più esperti la sfida di trovare una soluzione migliore. Per quelli meno esperti, sotto con i joysticks!

GEY

DESCRIZIONE DEL PROGRAMMA

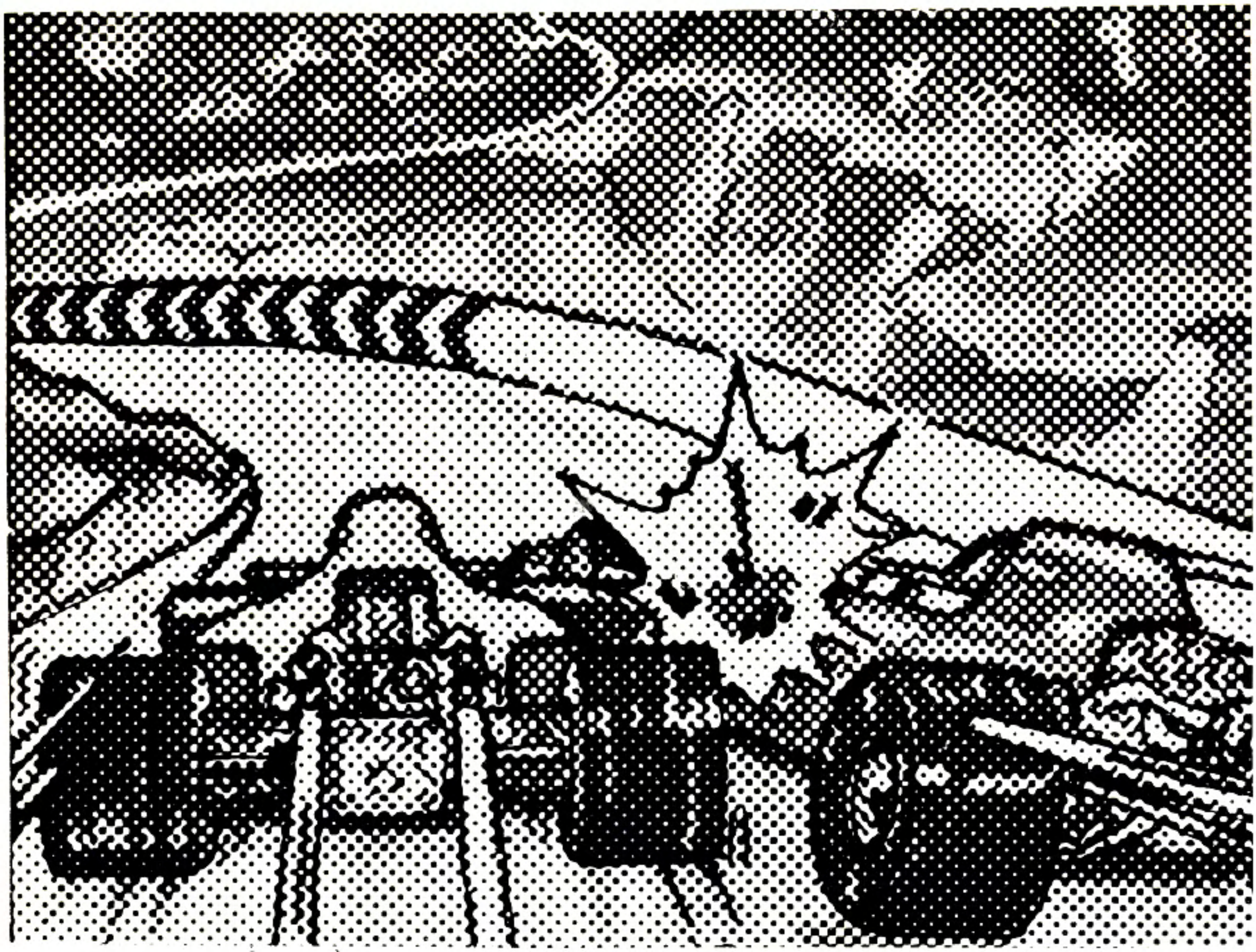
Linee	Descrizione della funzione
30- 40	Inizializzazione
50- 60	Schermo per inserzione dati
100-139	Definizione delle figure e dello sfondo
144-155	Posizionamento iniziale degii sprite
170-220	Routine principale
300-350	Determinazione degli sprite coincidenti
360-630	Azione susseguente alla coincidenza
700-822	Scambio giocatori e ripresa del gioco

GRAND PRIX

Le esaltanti vittorie dei più famosi piloti di formula uno potranno essere rivissute sul piccolo schermo del vostro video.
Basta infatti caricare con un OLD CS1 il programma Grand Prix per dare il via al vostro personale campionato mondiale di automobilismo.
Una volta lanciato col sempiterno RUN, il programma mostrerà su sfondo nero, che rimarrà per tutto il resto del gioco, la pagina di presentazione. Appariranno quindi i piloti vostri avversari con le loro vetture di colore blu, bianco, giallo, e verde (mentre la vostra sarà rossa) e la richiesta di battere un tasto per continuare. A operazione avvenuta apparirà una F gialla che vi permetterà di aggiudicarvi durante il percorso cinquanta litri di carburante extra e 100 punti.
Tempo pochi secondi e sarete lanciati su un lungo rettilineo dove potrete iniziare a tutta velocità il sorpasso delle vetture concorrenti cercando però di non urtarle e di non schiantarvi contro i verdi alberi che incontrerete. Non abbandonate mai il circuito altrimenti vedrete calare rapidamente la vostra riserva di carburante e di punti fino a giungere all'esplosione della vettura. Se sarete veramente in gamba riuscirete dopo quindici giri a giungere alla fine del percorso per ricevere i meritati onori e riposare le vostre stanche braccia. Per chi, a questo proposito, preferisse accorciare la lunghezza del tracciato (magari solo per vedere come va a finire) si consiglia di correggere il numero dei giri alla linea 750.
Al via disporrete di centocinquanta litri di benzina che andrà via via esaurendo incrementando però il vostro bottino di dieci punti. Un grosso boato e il bagliore di un esplosione segnaleranno che la vostra auto ha urtato contro qualcosa o che è finito il carburante, la vettura sarà prontamente sostituita.
Per muoversi sul precorso

REMARKS

100- 220	Assegnazione colori
230- 390	Assegnazione caratteri
400- 430	Presentazione
440- 450	CALL KEY d'attesa
460- 490	Riassegnazione di alcuni caratteri
500- 510	Ciclo temporizzatore
520- 570	Assegnazione variabili principali
580-1330	Percorso principale con display notizie (carburante e punti)
1340-1370	Scelta percorsi alternativi
1380-1500	Scelta costruzione ostacoli sulla pista
1510-1540	Aggiornamento variabili e controllo quantità carburante
1550-1590	CALL KEY di movimento
1600-1620	CALL JOYST di movimento
1630-1700	Controllo se colpisce auto o albero (dritto)
1710-1790	Controllo se colpisce auto o albero (sinistra) e spostamento
1800-1880	Controllo se colpisce auto o albero (destra) e spostamento
1890-2610	Primo percorso
2620-3060	Secondo percorso
3070-3120	Distruzione auto
3130-3150	GAME OVER e richiesta altra partita
3160-3190	CALL KEY finale
3200-3240	Sconfinamento dal tracciato
3250-3310	Controllo rifornimento benzina
3320-3520	Percorso finale di vittoria



delimitato dalle strisce bianco-rosse potrete usufruire della tastiera con i tasti di direzione S per la sinistra e D per la destra o dei joystick (1). Nel caso che non riusciate a completare la gara il computer, visualizzando il definitivo GAME OVER, aspetterà che voi gli comunichiate con una S o una N la vostra decisione di rifarvi o meno con un'altra partita. A questo punto innestate... la marcia e via sulla tastiera!

MARCO SQUINTANI

LISTA VARIABILI

T = Variabile di tempo
K = Contiene il codice ASCII del carattere premuto da tastiera
X = Variabile di scelta casuale del percorso
Z = Contiene il carattere letto dallo schermo nelle GCHAR
W = Variabile di informazione per l'uso del joystick
A$ = Contiene la stringa per la definizione caratteri
FL = Controllo del carburante
RI = Riga di visualizzazione della macchina del giocatore
CO = Colonna di visualizzazione della macchina del giocatore
TI = Variabile di tempo
KEY = Contiene il carattere per la scelta di una nuova partita
GIR = Variabile contenente il numero dei giri effettuati
COL = Variabile di ciclo nelle CALL COLOR
PTS = Incremento dei punti
CAR = Controllo delle macchine ancora a disposizione
SCE = Variabile di scelta casuale per il piazzamento di ostacoli sulla pista

TI 99 MARKET

Finalmente software di altissimo livello ed assolutamente inedito per il Vostro TI 99/A4.
A sole L. 30.000 cadauno (10% di sconto agli abbonati) sono disponibili i seguenti programmi:

TI 99 CALC

Si tratta della riduzione per TI 99/4A di uno dei packages più diffusi sui personal professionali. Consente di produrre molto facilmente programmi capaci di risolvere qualsiasi problema riconducibile in forma tabellare. Contiene un vero e proprio nuovo linguaggio di programmazione facile da apprendere ed estremamente potente. Consente di memorizzare dati (numeri o stringhe di caratteri) in forma tabellare incasellandoli in elementi di tabella contraddistinti da un numero di riga (01, 02, ...) e da una lettera che identifica la colonna (A, B, C, ...). Consente anche di associare formule agli elementi di tabella. In base a tali formule verranno calcolati i valori degli elementi di tabella associati. *Documentazione:* Viene fornita una completissima documentazione ed istruzione per l'uso. *Equipaggiamento necessario:* EXTENDED BASIC, registratore a cassette, espansione di memoria 32 K (opzionale).

1000 NOMI

Un gruppo di amici davanti al video fornisce il proprio nome al computer. Il computer sceglie una lettera dell'alfabeto, e interroga i giocatori a turno, richiedendo che venga pronunciato, in 5 secondi, un nome proprio (o un nome di città, o di animale, o di un altro genere, a seconda degli accordi iniziali) che cominci per la lettera prescelta. Non valgono le ripetizioni. Chi accumula un numero di errori prestabilito, viene eliminato. Per esclusioni successive, rimane un vincitore. *Caratteristiche:* Gioco di società richiedente riflessi rapidi e memoria. Voce in italiano con leggero accento inglese. Scritto in BASIC. *Documentazione:* Il programma contiene le istruzioni per l'uso. *Equipaggiamento necessario:* Modulo Terminal Emulator II, Speech Synthesiser, registratore a cassette.

ELEUSIS

Il computer pensa una sequenza segreta di carte francesi (per esempio: una carta rossa, una di picche, e una dispari; oppure, una carta di cuori e una carta di valore superiore alla precedente). I giocatori (fino ad 8) hanno a disposizione inizialmente 12 carte ciascuno. A turno, provano a giocare una carta. Se essa si inserisce correttamente nella sequenza, il computer la dispone nella fila principale sullo schermo, altrimenti in una fila secondaria. Il giocatore che ha piazzato una carta accettata dal computer, ha diritto di giocare ancora. Vince chi, indovinando per primo la sequenza, riesce a liberarsi di tutte le sue carte piazzandole nella fila principale oppure dimostra che non possiede più carte per la fila principale. Il gioco si esegue con 4 livelli di difficoltà e con l'opzione di scommettere su ogni carta giocata. *Documentazione:* ELEUSIS viene fornito con le istruzioni dettagliate in italiano. *Equipaggiamento necessario:* 32K Memory Expansion, Modulo Extended Basic, registratore a cassette.

GIROTONDO

Un gruppo di amici siede davanti al video, formando un semicerchio. Ognuno deve pronunziare a turno, velocemente, un numero progressivo da 1 a 99. Ma l computer dà, per ogni nuova tornata, istruzioni diverse: per esempio, i numeri contenenti il 2 e il 9 non vanno pronunciati, ma bisogna che al suo turno l giocatore alzi le mani se il numero contiene un 9; che fischi se contiene un 2. Chi sbaglia, paga un pegno. Anche il computer partecipa al gioco, facendo parte del semicerchio. Ore di divertimento assicurate con amici. *Caratteristiche:* Gioco di società richiedente riflessi rapidi. Voce in italiano (con leggero accento inglese). Grafica. Scritto in BASIC. *Documentazione:* fornito completo di 2 versioni: "GIROTONDO 1" pe radulti, e "GIROTONDO 2" per bambini. Completo di istruzioni per l'uso. *Equipaggiamento necessario:* Modulo Terminal Emulator II, Speech Synthesiser, registratore a cassette.

TOMBOLA

La tombola al computer, con voce in italiano e napoletano. Interpretazione dei numeri secondo la classica smorfia. Ripartizione iniziale della posta e verifica dei numeri usciti. *Caratteristiche:* Gioco di società di pura fortuna. Voce con leggero accento inglese. Cartelle e segnanumeri per i giocatori non compresi. Scritto in BASIC. *Documentazione:* Il programma contiene le istruzioni per l'uso in italiano. *Equipaggiamento necessario:* Modulo Terminal Emulator II, Speech synthesiser, 32K memory expansion, registratore a cassette.

SCOPA

Il classico gioco della Scopa: lo sfidante contro il computer. Tutter le regole della Scopa classica, con rappresentazione grafica delle carte napoletane, e visualizzazione alla fine di ogni mano del punteggio ottenuto. Il gioco è disponibile in due versioni: con e senza Speech Synthesizer. *Documentazione:* SCOPA viene fornito con le istruzioni dettagliate in italiano. *Equipaggiamento necessario:* 32K Memory Expansion, Disk Drive, Modulo Extended Basic, Joystick, Speech Synthesizer opzionale.

QUADRO X

Il gioco si compone di due fasi.
Fase 1: L'utente dipinge un quadro sullo schermo utilizzando mosaici colorati elementari di varia forma preventivamente scelti. A quadro finito, il risultato viene immagazzinato su disco. E' possibile preparare più quadri, uno dopo l'altro, da utilizzare per la seconda fase.
Fase 2: Un gruppo di persone davanti al video, vedrà pian piano ricomparire i mosaici fino al quadro completo. Il primo che indovina l'oggetto del quadro vince. Si possono così organizzare divertenti sedute a quiz con amici. *Caratteristiche:* Gioco di inventiva artistica e di società, richiedente perspicacia. Per la composizione del quadro, sono disponibili 16 mosaici di forma diversa e 6 coppie di colori. *Documentazione:* Il programma contiene le istruzioni per l'uso e viene fornito completo di 2 versioni: «QUADRO X1», per la fase 1 e «QUADRO X2» per la fase 2. *Equipaggiamento necessario:* 32K Memory Expansion, Modulo Extended Basic, Disk.

SNOOPY

Snoopy si muove su una piattaforma di caselle esagonali vista in prospettiva. Man mano che Snoopy avanza, gli esagoni si illuminano mostrando il percorso. In una delle caselle è nascosta una scala per discendere dalla piattaforma; in un'altra, il percorso per raggiungere la sua cuccia. Ma attenzione! Snoopy potrebbe malamente cadere dalla piattaforma. Inoltre in alcune caselle si nascondono trappole pericolose, come palle da baseball. palle da football, cerchi magici, botole, zucche intagliate, molle rimbalzanti. In una delle caselle, il percorso finora compiuto viene spento. Vince il giocatore che manovra i suoi Snoopy per fare più punti attraverso tre schermi di colori diversi. I punteggi record vengono memorizzati su disco. *Documentazione:* SNOOPY viene fornito con le istruzioni dettagliate in italiano. *Equipaggiamento necessario:* 32K Memory Expansion, Disk Drive, Modulo Extended Basic, Joystick.

2000 COWBOY

Un intraprendente cowboy, alla ricerca di dollari, si trova intrappolato in una stanza. La stanza ha 3 porte chiuse sul fondo. Il cowboy sceglie quale porta aprire. La porta si apre... e una figura appare sulla soglia. Pochi attimi per decidere se sparare o meno. La figura potrebbe essere amica: gatto, pistola, segnale, aiutante, robot, dollari, bomba; o nemica: diavolo, mostro, missile, anti-robot, fantasma, piovra, killer, ragno, buco nero, serpente. Alcune conviene spararle, (se ci sono colpi in canna!) altre no. Altre invadono la stanza pericolosamente. Altre uccidono senza scampo. Il robot è un utile amico: seguirà fedelmente il suo padrone, mostrandogli cosa si nasconde dietro una delle porte. Se il cowboy si salva, passa attraverso la porta aperta per penetrare in una successiva stanza simile. Finchè non trova i dollari. Tanti dollari, se ci sa fare. E poi la bomba, per far esplodere l'allucinante serie di stanze, e riacquistare la libertà con il bottino. *Caratteristiche:* Gioco richiedente riflessi, decisioni rapide, azione, e un pizzico di fortuna. Eccellente grafica tridimensionale. Al livello dei più sofisticati Intellivision/Atari. Espansione di memoria totalmente occupata. *Documentazione:* Il programma viene fornito in 2 versioni: "2001 COWBOY", richiedente Speech Synthesiser con voce in inglese; e "2002 COWBOY", di esecuzione più rapida e senza voce. E' fornito inoltre a corredo "2000 REGOLE", con le regole del gioco in italiano e grafica. *Equipaggiamento necessario:* 32K Memory Expansion, Disk Drive, Modulo Extended Basic, Joystick, Monitor/TV a colori, Speech Synthesiser (opzionale).

RADAR

Le formazioni nemiche si schierano sul teatro della battaglia. Il giocatore ha a disposizione cannoni, elicotteri, mine, raggi laser, e missili. Il nemico, aerei e carri armati. Il giocatore sceglie, di volta in volta, strategicamente l'arma con la quale intende colpire, e istruisce il computer per l'attacco o spara direttamente. Ma attenzione! Per ogni serie di colpi a vuoto, a seconda del livello di difficoltà e del punteggio, il nemico contrattacca e distrugge senza scampo. Il giocatore accumula punti, ma la battaglia svolge inevitabilmente a suo sfavore. Finchè... non viene attivata l'arma segreta. *Caratteristiche:* Gioco combinato di strategia e di riflessi. Ottima grafica multisprite ad alta definizione. Complessità al livello dei più sofisticati Intellivision/Atari. Espansione di memoria 32K totalmente occupati. *Documentazione:* RADAR viene fornito corredato con un'appendice di supporto "REDAREGOLE" che contiene le istruzioni dettagliate in italiano. *Equipaggiamento necessario:* 32K Memory Expansion, Disk Drive, Modulo Extended Basic.

TI99 NEWSOFT

DICEMBRE 1984 L. 10.000

EDISOFT

FINALMENTE IL MERGE SU CASSETTA

BASIC: I file su cassetta

ASSEMBLER: Il modo TEXT

IN SOFTWARE & SOFTWARE:
Merge - Note di NATALE -
Torre di Hanoi - Star Wars...

SOMMARIO

□ **COMINCIAMO DA ZERO**
Questa volta per la rubrica del BASIC parliamo della gestione dei Files su cassetta — pag. 3-4

□ **EXTENDED**
Il multistatement forzato, ovvero come fare per fare accettare al computer righe più lunghe di 140 caratteri — pag. 5-6

□ **ASSEMBLER**
Il modo TEXT — pag. 7

□ **AGENDA TELEFONICA**
Un programmino in BASIC per crearsi una rubrica facile da consultare — pag. 8-9

□ **STATISTICA**
Come confrontare due grandi medie — pag. 10-11

□ **STAR WARS**
Un videogame avvincente e pieno di suspence. Gira in Extended — pag. 12-13

□ **NOTE DI NATALE**
Il gioco Simon in versione natalizia — pag. 14

□ **DIDATTICA**
Anche se siamo a Natale, non dimentichiamoci che con il computer si può anche studiare — pag. 15-17

□ **TORRE DI HANOI**
Un gioco di strategia e una sfida aperta fra tutti i lettori — pag. 18

□ **NATALE SU TI 99/4A**
Con questo programma apriamo la cassetta di questo numero di Dicembre. Auguri! — pag. 19

□ **DONI CELESTI**
Povero Babbo Natale. Fate si che non gli cadano i pacchi che deve consegnare, gira in Basic — pag. 20-21

□ **SCROLL**
E' un programma per la MINI MEMORY e consente di creare sfondi in movimento — pag. 22

□ **MERGE**
Se avete l'EXTENDED BASIC e l'espansione di memoria da 32 K, questo programma vi consente di "fondere" due programmi su cassetta per una lunghezza complessiva di ca. 20 K — pag. 23-24

□ **INVENZIONI** — pag. 25

□ **LIBRI** — pag. 25

□ **LA POSTA** — pag. 26-27

□ **PICCOLI ANNUNCI** — pag. 28-29

□ **LA VIGNETTA** — pag. 30

□ **COLLABORATE CON NOI** — pag. 31-32

TI 99 NEWSOFT

ANNO 1 - Numero quattro
DICEMBRE 1984
MENSILE
Direttore responsabile:
ANTONIO C. LOSITO
Coordinatore Editoriale:
VINICIO CIVININI
Art Director:
BEPPE RE-FRASCHINI
Illustrazioni:
MARIA TERESA COCOZZA
Segreteria:
EMANUELA CHIAPELLO
CRISTINA FRISON
Redazione:
FIORENZA AURIEMMA
DARIO BAREZZI
Collaboratori:
SERGIO BORSANI,
GIANPAOLO BOTTIN,
DANIELE CATALFAMO,
PAOLO CIVARDI,
STEFANO DELLI PONTI,
FRANCESCO MACALI,
MAURO MATTIAZZI,
EZIO MONTINI,
LEVIO PEROTTI,
RICCARDO ROSSI,
MARCO SQUINTANI,
ALBERTO STRAFILE,
SEBASTIANO TOMASELLO,
PAOLO VENTAFRIDDA
TI 99 NEWSOFT è una pubblicazione della **NEWSOFT snc**,
via Stefano Jacini 4 - 20121 MI
Tel. 02/807464 - Telex 324284
REGISTRAZIONE AL TRIBUNALE DI MILANO N. 323 DEL 7.7.84

COPYRIGHT NEWSOFT snc
TUTTI I DIRITTI RISERVATI
Manoscritti e foto originali, anche se non pubblicati, non si restituiscono ed è vietata la riproduzione anche parziale di testi, fotografie e listati.

Pubblicità: AV STUDIO
Via Giuseppe Meda, 23
Tel. 02/8357155
Abbonamenti: l'abbonamento annuo a 11 numeri di *TI 99 NEWSOFT* costa L. 95.000 per l'Italia; L. 160.000 per l'Europa (spedizione aerea); L. 200.000 per tutti gli altri paesi. L'importo, per assegno non trasferibile o per vaglia postale, va inviato a NEWSOFT snc, via S. Jacini 4 - 20121 Milano.
Stampa: CGB, Albegno di Treviolo (BG)
Concessionaria per la distribuzione:
EUROSTAMPA Srl - C.so Vittorio Emanuele 111 - 10123 Torino - Tel. 011/538166-7

L'editore non si assume responsabilità alcuna sul contenuto degli articoli e/o software inviati. Per gli articoli firmati e/o siglati da collaboratori esterni, la redazione si assume la responsabilità prevista dalle leggi sulla stampa.

Tutti possono collaborare a TI 99 NEWSOFT, sia con articoli che con programmi. Scriveteci o telefonateci in redazione per avere tutti i ragguagli.

C'ERA UNA VOLTA...

Così potrebbe incominciare una favola, e questa favola potrebbe essere quella del TI 99, un piccolo-grande computer dimenticato e poi...
Invece questa volta incominciamo con C'È.
C'è il TI 99 e chi ancora ci crede.
Ad un anno dalla sua «soppressione» (vi ricordate il Natale dello scorso anno? Il TI 99 veniva svenduto per poche lire), si è acceso un movimento silenzioso di utenti — come ben dice il dott. Giorgio Massazza in una lettera — che hanno dichiarato apertamente di non condividere la decisione della casa produttrice di «abbandonare» il piccolo TI.
Bene! È proprio grazie alla fiducia e costanza di tutti gli utenti del TI, e non ultimo alle notevoli qualità della «macchina» molte delle quali ancora da scoprire, che da questo mese TI 99 MARKET incomincia ad offrire hardware e software di vario tipo, per completare e «personalizzare» il nostro computer.
Adesso se andate in fondo alla rivista vedrete che finalmente c'è l'introvabile Extended Basic, il Logo, espansioni, e così via.
È vero, mancano periferiche come: il controller per i drive, la tavoletta grafica o la penna ottica e l'espansione RAM da 198 K... Ma se avete un po' di pazienza!
Poi avete notato i prezzi?
E i prezzi per chi è abbonato alla rivista?
A proposito: sinceri ***auguri*** *a tutti voi dalla redazione di TI 99 NEWSOFT - la Vostra rivista.*

GESTIONE FILES SU CASSETTA

La gestione di files si rende indispensabile in quei casi dove si presenta la necessità di rielaborare più volte gli stessi dati, senza pertanto inserirli da tastiera ogni volta che se ne deve fare uso. Inoltre i files possono essere modificati con estrema facilità e velocità, fornendo una migliore soluzione del comune metodo di immagazzinare i dati in linee di data.
Esaminiamo ora le varie operazioni per l'apertura e la gestione dei files.
Per prima cosa dobbiamo predisporre il computer a comunicare con un dispositivo esterno di memorizzazione, nel nostro caso il registratore, che ha come parola chiave CS1. Preferiamo utilizzare l'unità CS1 perché in grado di effettuare entrambe le operazioni di scrittura e lettura.
L'istruzione OPEN è incaricata di aprire il canale di comunicazione. La sintassi è:

OPEN= numero file: nome file[,organizzazione file] [,tipo file] [,modo di apertura] [,tipo record]

— NUMERO FILE.
Il numero del file è un numero intero compreso tra 1 e 255.
Questo numero serve per distinguere i file nel caso si presenti la necessità di aprirne più di uno contemporaneamente. Tuttavia il computer non può aprire più di tre files nello stesso programma.
Il numero deve essere preceduto dal segno = «cancellato».
E' possibile sostituire con una variabile il numero, ma essa deve contenere un numero intero compreso tra 1 e 255.

— NOME FILE.
Il nome del file deve essere posto tra virgolette e deve contenere il suffisso (es. «CS1» o «RS232») della periferica con cui si desidera comunicare.
Trattandosi di una stringa è possibile sostituirla con una variabile, ovviamente di stringa, purché contenga il nome esatto della periferica.

— ORGANIZZAZIONE FILE.
I records contenuti nel file possono essere registrati in due diversi modi: sequenzialmente, specificando SEQUENTIAL o accesso diretto specificando RELATIVE.
Utilizzando il registratore a cassette, possiamo organizzare il file solo in modo sequenziale, perché detta unità non può avere un accesso diretto al singolo records, cosa possibile con una unità a di-

SCHEMA DI APERTURA FILE SU CS1

OPEN **à** numero file: nome file[,organizzazione file] [,tipo file] [,modo di apertura] [,tipo record]

— Numero file	numero intero tra 1 e 255
Nome file	CS1 o CS2
— Organizzazione file	SEQUENTIAL
— Tipo file	INTERNAL o DISPLAY
— Modo di apertura	INPUT «lettura»
— Modo di apertura	OUTPUT «scrittura»
— Tipo di record	FIXED 64
— Tipo di record	FIXED 128
— Tipo di record	FIXED 192

L'unità CS1 può aprire file di tipo OUTPUT o INPUT.
L'unità CS2 può aprire file solo di tipo OUTPUT.

Figura 1

sco.
La specifica SEQUENTIAL può essere omessa, perché viene sottintesa dal computer.

— TIPO DI FILE.
Il computer può trasmettere e ricevere dati codificati in due modi:

a) codificati in forma binaria, specificando INTERNAL,
b) codificati sotto forma di codici ASCII, specificando DISPLAY.

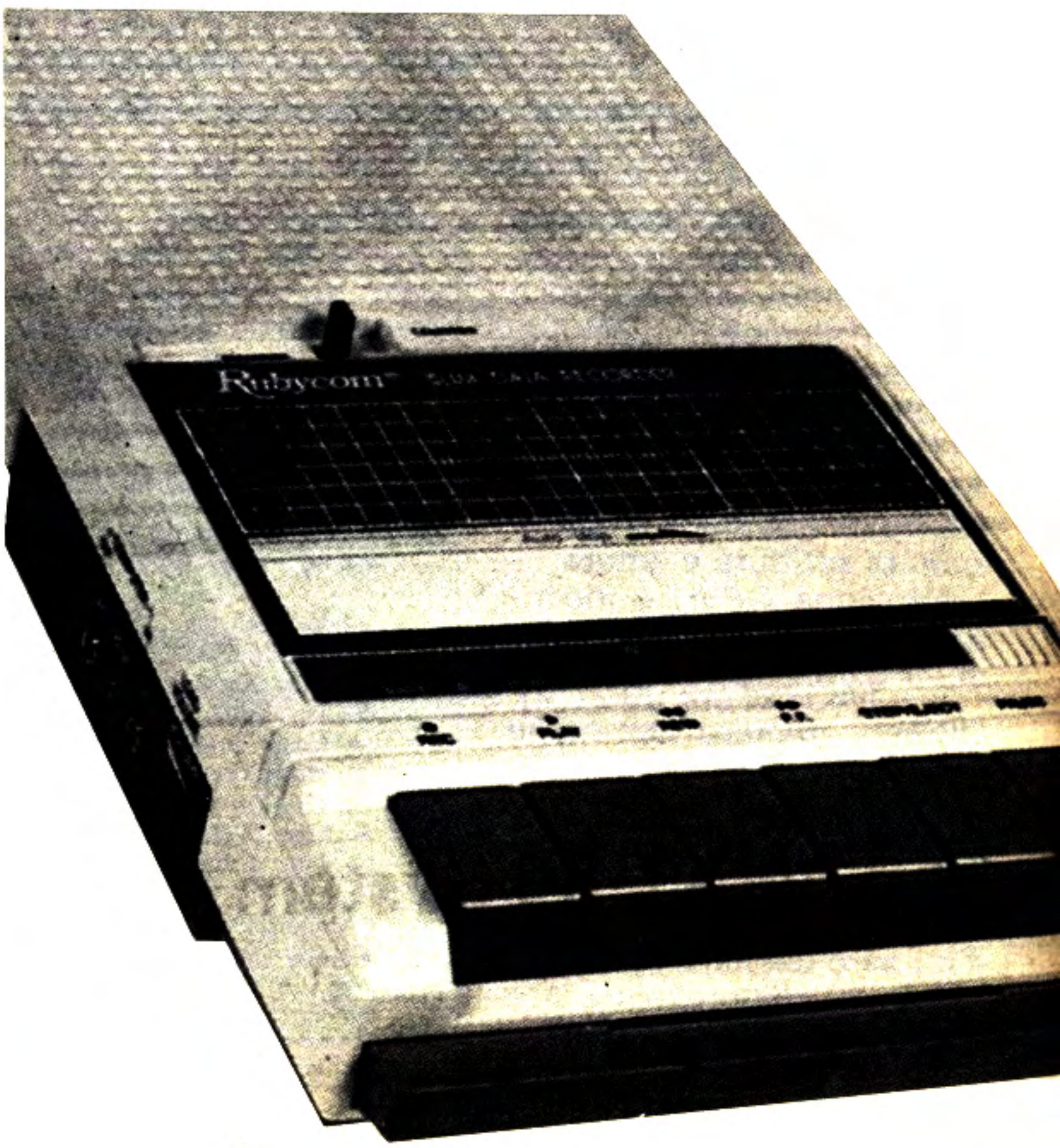
Il modo INTERNAL permette alla macchina di poter lavorare con maggiore velocità perché i dati binari sono direttamente interpretabili dal computer e non sono soggetti a ulteriori codificazioni.
Per codificazione si intende la trasformazione di dati sotto forma di codici ASCII in dati binari, le uniche informazioni che il computer può capire e quindi eseguire.
L'utilizzo dell'opzione DISPLAY è quasi sempre sconsigliata perché i codici ASCII prodotti sono soggetti a codifica, ed essa richiede un determinato tempo.

— MODO DI APERTURA
Con l'unità a cassette, possiamo aprire il file in due modi:
a) in modo scrittura, specificando OUTPUT, abilitando il computer a trasmettere i dati all'unità esterna,
b) in modo lettura, specificando INPUT, abilitando il computer a ricevere i dati dall'unità periferica.

— TIPO DI RECORD.
Il record può essere di lunghezza fissa o variabile, nel primo caso dovremo specificare la parola FIXED, seguita da un numero che rappresenta la lunghezza massima complessiva di tutti i campi del records.
Nel registratore a cassette abbiamo tre diverse possibilità:
records della lunghezza di 64, 128 o 192 caratteri.
Se dopo la parola FIXED non viene specificato alcun valore, verrà assegnato automaticamente il valore di 64.
Nel caso il record non raggiunga la lunghezza specificata, verrà completato con codici ASCII 32, se avremo aperto un file DISPLAY, e con 0 binari nel caso sia stato aperto in modo INTERNAL.
In contrapposizione troviamo la specifica VARIABILE.
Essa, pur mantenendo le stesse caratteristiche generali di FIXED «cioè seguita da un numero ecc.», non completa il record ne con codici ASCII 32 ne con 0 binari. La specifica VARIABILE può essere utilizzata solo con le unità a dischi.
Nella figura 1 potrete trovare uno schema sintetico di tutto quello che è stato spiegato in questa prima parte dell'articolo.
Nella cassetta troverete un programma di gestione file «AGENDA TELEFONICA», che mette in pratica quello che abbiamo detto e ciò che diremo nel prossimo articolo.

EZIO MONTINI

(continua)



MULTISTATEMENT FORZATO

Ad alcuni di voi sarà certamente capitato di vedere apparire il malefico messaggio di MEMORY FULL, dopo aver elaborato programmi di considerevoli dimensioni. Le cause che possono provocare tale messaggio possono essere molte, una di queste può esser l'eccessiva lunghezza del programma.
In extended basic esiste un modo per risparmiare memoria, e quindi ci permette di scrivere programmi con lunghezze ben superiori a quelle normalmente consentite, forse dal titolo avrete capito che si tratta di un metodo per scrivere più istruzioni su una stessa riga.
Questa possibilità di multistatement non vi suonerà di certo come una novità, ma quanti di voi sanno che è possibile scrivere più di cinque righe di istruzioni?
Il computer normalmente accetta righe di lunghezza non superiore ai 140 caratteri, ma adottando un particolare accorgimento è tuttavia possibile superare tale limite.
Le operazioni che dobbiamo effettuare sono:
a) scrivere normalmente la nostra riga di istruzioni, ovviamente separandole con il segno convenzionale "::", fino al raggiungimento della quinta riga;
b) premere ENTER;
c) premere FCTN 8 (corrispondenti a REDO);
d) portarsi con il cursore all'ultima istruzione inserita, poi continuare a scrivere (le istruzioni) fino al raggiungimento delle nove righe.
ATTENZIONE: vi sono dei limiti, ad esempio non è possibile generare più di quattro sprites su una stessa riga, non è consentito di inserire più di cinque CALL (HCHAR, VCHAR ecc.) su un'unica riga.
Questi limiti sono fissati dalla macchina e sono stati scoperti dopo parecchi tentativi.
Il limite massimo consentito di altre istruzioni non è ancora stato stabilito, sappiamo solo che è consentito un maggior numero di assegnazioni di variabili. Se, una volta scritte le fatidiche nove righe di istruzioni, il computer non ci segnala il noto errore «LINE TOO LONG», è possibile a volte aumentare la lunghezza della linea fino a dieci o undici linee, operando come se si trattasse di una normale linea di istruzioni, applicando il metodo sopra descritto.
Questo metodo permette di sfruttare al massimo la memoria disponibile e consente ai programmatori del «MEMORY FULL» di non abbreviare il programma «ampu-tandone» delle parti.
Un altro motivo per cui è più conveniente applicare questo sistema è che il computer esegue il programma in maniera leggermente più veloce.

RUN AUTOMATICO DI PROGRAMMI

Il titolo è certamente molto azzardato, ma vedrete che non rimarrete ugualmente delusi.
Si tratta infatti di una concatenazione di programmi, cioè chiamati uno di seguito all'altro.
L'istruzione che ci permette di eseguire il concatenamento è RUN "CS1".
Quando il programma incontra un RUN riferito a dispositivi periferici, mette in esecuzione la routine in linguaggio macchina, (in realtà si tratta di GPL per accontentare i più esperti), di caricamento di dati, riferita al dispositivo indicato tra le virgolette, nel nostro caso CS1.
Una volta caricato, lo manda in esecuzione. Questa possibilità può essere sfruttata in vari modi, uno tra i più interessanti è quella di poter predefinire dei patterns.
Solitamente il nostro computer cancella i patterns sino a

quel momento definiti se il programma viene fermato, lasciando ridefiniti solamente i codici da 96 a 143.
Utilizzando il metodo sotto indicato, risolveremo tutti questi problemi, riuscendo a mantenere invariati tutti i patterns da noi definiti compresi i codici da 32 a 95.
La parte più pesante di un programma di gioco è senza dubbio quella di definizione dei caratteri, infatti essa appesantisce il programma, riducendone la velocità di partenza.
I patterns dei caratteri vengono caricati nella memoria VDP del computer, una buona idea sarebbe caricare i patterns nella memoria con un programma a parte, permettendo al programma principale di poter usufruire della memoria in maniera migliore.
Noi vi proponiamo il metodo per attuare tale idea.
Si tratta di un piccolo programma esplicativo che potrà essere modificato facilmente a seconda delle vostre esigenze.
Il trucco è di porre l'istruzione RUN «CS1» dopo la definizione dei caratteri. Infatti il computer dopo aver definito le forme, rimarrà sempre in modo esecutivo, eliminando l'inconveniente della cancellazione dei patterns ridefiniti. Tali caratteri potranno essere usati dal programma che verrà caricato e mandato in esecuzione successivamente.

Il programma che li dovrà utilizzare deve avere alcune caratteristiche:

1) Non deve interrompere la sua esecuzione, perché ciò comporterebbe la ridefinizione dei caratteri da 32 a 96;

2) Non deve contenere eventuali ridefinizioni dei caratteri già definiti in precedenza, se non volete cambiare la forma del pattern con una da voi scelta.

IL PROGRAMMA

Scrivete i programmi di figura 1 e salvateli separatamente su cassetta, prima il programma A, poi il B.
Cancellate la memoria con NEW; caricate il programma A e mandatelo in run, quando avrà finito di ridefinire i caratteri, posizionate il nastro sul programma B, e quando vi sarà richiesto, caricatelo.
In pratica il programma A contiene la ridefinizione delle maiuscole, e il B invece le utilizza.

EZIO MONTINI

```
100 REM * * * * * * * * * * * * * * * * * * * * * * *
110 REM * Caratteri in corsivo
120 REM * BY
130 REM * Montini Ezio
140 REM *
150 REM * e caricamento in VDP
160 REM * Programma A:
170 REM * * * * * * * * * * * * * * * * * * * * * * *
180 DATA 000000384848340O, 004040704848300O,
    000000384040380O, 000808384848340O,
    000018243C201800, 001828203820200O
190 DATA 000038484838083O, 002020382424240O,
    001000301010380O, 001000301010502O,
    004040506050480O, 003010101010380O
200 DATA 0000006C54545400, 000000784848480O,
    000000304848300O, 000705848704040,
    000038484838080B, 000000283020200O
210 DATA 000018203808300O, 001010381010180O,
    000000484848340O, 000000242428100O,
    000004454542Bùù, 000000281028280O
220 DATA 000000484830102O, 000000381020380O
230 RESTORE 180
240 FOR ZO = 97 TO 122
250 READ Z $
260 CALL CHAR (ZO, Z$)
270 NEXT ZO
280 FOR ZO = 33 TO 96 : : CALL CHARPAT (ZO, Z $) : : Z $ = SEG
    $ (Z $, 3, 14) & "00" : : CALL CHAR (ZO, Z $) : : NEXT ZO
290 FOR ZO = 123 TO 126 : : CALL CHARPAT (ZO, Z $) : : Z $ =
    SEG $ (Z $, 3, 14) & "00" : : CALL CHAR (ZO, Z $) : : NEXT ZO
300 RUN "CS1"

100 CALL CLEAR
110 REM * * * * * * * * * * * * * * * * * * * * * * *
120 REM * Programma : B *
130 REM * * * * * * * * * * * * * * * * * * * * * * *
140 FOR A = 32 TO 149
150 PRINT CHR $ (A);
160 NEXT A
170 GOTO 130
```

PRIMI PASSI
CON L'ASSEMBLER DEL TI-99/4A

Eccoci giunti al terzo appuntamento con la nostra rubrica dedicata all'assembler, ma prima di continuare, dobbiamo ringraziare tutti i lettori che ci scrivono o telefonano. Le richieste e le domande relative all'assembler sono veramente tante, e ciò non può che farci piacere.
Continuiamo dunque con il nostro discorso.

TEXT

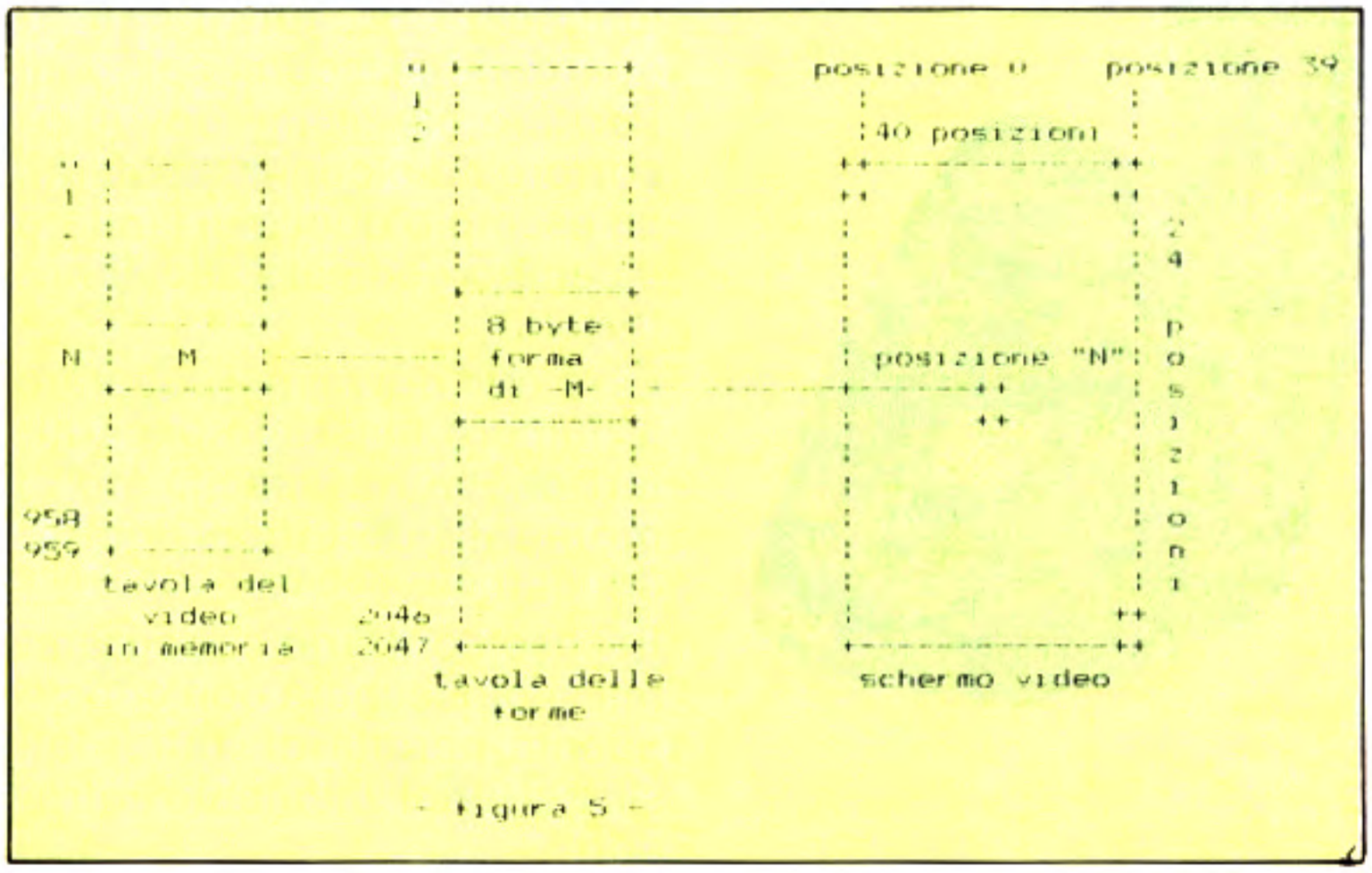
Il <TEXT MODE>, come dice la parola, «modo testo», viene impiegato sostanzialmente per la visualizzazione di testi (world processing, listati, ecc.), trasformando così il video in una specie di foglio elettronico. Certamente è possibile utilizzare anche i modi grafici per la visualizzazione di testi, ma con il «TEXT MODE» si ha il vantaggio di lavorare con 40 colonne tutte visibili sullo schermo.
Di contro, il «TEXT MODE» è poco adatto per l'utilizzo in giochi o animazioni, in quanto non è possibile utilizzare gli sprites, e la scelta del colore, è unica per tutti i caratteri.

La tavola dei colori, che abbiamo visto nei modi grafici, in questo caso, non esiste. Il registro n. 7 «VDP», contiene da solo tutte le informazioni relative al colore, sotto forma di due codici di quattro bit, corrispondenti al colore del carattere e al colore dello sfondo. (figura 4).

- figura 4 -

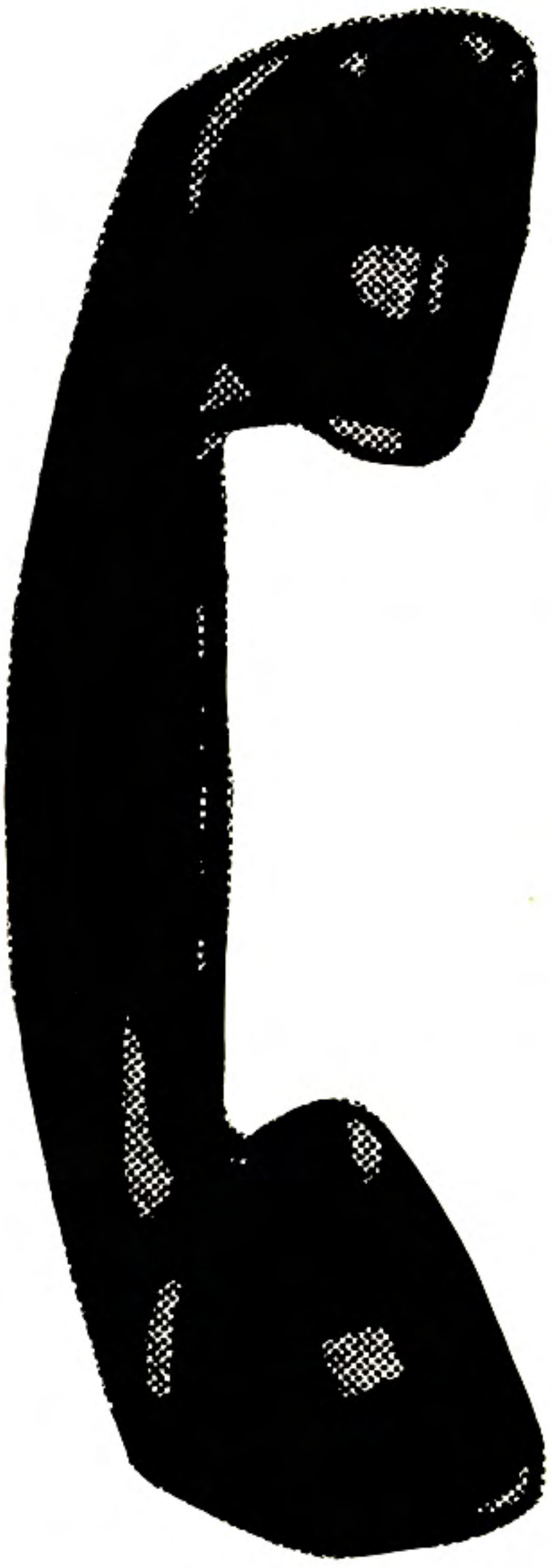
La tavola delle forme, è identica a quella che abbiamo già visto in «GRAPHIC1» occupa quindi 2048 byte, contiene 256 forme da 8 byte ciascuna, che numerate da 0 a 255 corrispondono ai nomi dei caratteri. In questo caso però la tavola del video occupa 960 byte, in quanto conserva un byte per ogni posizione carattere su video (40x24=960), ricordiamo inoltre che la tavola del video, contiene i nomi dei caratteri (0-255). La mappa del modo testo (figura 5), vi chiarirà meglio le idee.
SEBASTIANO TOMASELLO

(3 - continua)



- figura 5 -

AGENDA TELEFONICA TI BASIC



Questo programma è stato creato appositamente da noi, per soddisfare tutte le persone che ci hanno scritto, richiedendoci di affiancare a lezioni teoriche degli esempi pratici di immediato utilizzo e di qualche utilità pratica.
Agenda telefonica è a nostro giudizio un ottimo esempio di gestione di file.
Il programma genera un file sull'unità CS1 che contiene tutti i dati della nostra agenda personale.
Per sfruttare al meglio le possibilità del registratore, e per una maggiore velocità, di assunzione e trasmissione dei dati, abbiamo pensato di compattare in solo record, di lunghezza 64, tutti i dati riguardanti ciascuna persona.
Per fare ciò siamo dovuti ricorrere ad alcuni espedienti, ad esempio quello di limitare il campo dedicato alle annotazioni.
Esso non dovrà mai superare il numero massimo di caratteri che verrà specificato al momento dell'inserimento dei dati.
In caso contrario il programma si blocccherà con conseguente perdita di tutti i dati sino a quel momento introdotti.

IL PROGRAMMA

Il limite massimo di dati che si possono inserire è fissato a 100, ma è possibile portare tale numero sino ad un massimo di 300, modificando la riga numero 210, sostituendo al numero di dimensionamento delle matrici il numero voluto.
È consigliabile non immettere più di 100 dati per ogni volta perché il tempo impiegato per salvarli è di circa 10 minuti.
Per velocizzare questa operazioni si possono creare tanti files, quante saranno le lettere dell'alfabeto, caricando in macchina solo quello interessato.
(Es. il nome da ricercare è MARIO ROSSI, noi caricheremo il file che contiene i nomi che cominciano con R).
Questo metodo presenta dei lati negativi dato che il programma prevede ben tre criteri di ricerca:

— per cognome
— per numero telefonico
— per indirizzo

Per poter usufruire di tale possibilità dovremo caricare i dati di tutto l'archivio, cosa impossibile dato che abbiamo creato molti files.

AGENDA TELEFONICA

Linee	Descrizione della funzione
100- 250	Inizializzazione delle variabili, selezione dei colori
260- 500	Schermo iniziale e schermo selezione con scelta
510- 520	Smistamento alle routines scelte
530- 770	Procedura di inserimento dei dati
780- 890	Procedura per il salvataggio dei dati su CS1
900-1010	Procedura per il caricamento dei dati da CS1
1020-1200	Routine di ricerca dei dati per cognome
1210-1390	Routine di ricerca dei dati per numero telefonico
1400-1580	Routine di ricerca dei dati per l'indirizzo
1590-1660	Opzione di fine lavoro
1670-1840	Routine per la stampa dei dati ricercati

VARIABILI NUMERICHE

WE,W,K,S,Q .... Variabili di comodo
LE ..... Variabile per il controllo del default 64
R ..... Variabile contenente il numero di records inseriti

VARIABILI MATRICIALI DI STRINGA

C$ (100) ..... Matrice contenente i cognomi
IN$ (100) ..... Matrice contenente gli indirizzi
N$ (100) ..... Matrice contenente i nomi
NOT$ (100) ..... Matrice contenente le annotazioni
T$ (100) ..... Matrice contenente i numeri telefonici
TW$ (100) ..... Matrice contenente le città

VARIABILI DI STRINGA

CC$ ..... Variabile contenente il cognome da ricercare
IND$ ..... Variabile contenente l'indirizzo da ricercare
LU$ ..... Variabile contenente la lunghezza dei campi per il default 64
RI$, R1$ ..... Variabili di comodo
R2$, R3$ ..... Variabili di comodo
TT$ ..... Variabile contenente il numero telefonico da ricercare

Un'idea potrebbe essere: registrare dei files contenenti dati con diverse iniziali, sempre limitandosi nel fare files di lunghezza non superiore ai 100 record.
In fase di ricerca cominceremo a caricare il primo file, nel caso non sia presente il dato da noi richiesto, procederemo a caricare il secondo e così via.
Questa soluzione permette di sfruttare al massimo le possibilità di ricerca di questo programma.
A voi la scelta!

EZIO MONTINI

CONFRONTO TRA MEDIE

Nel corso della nostra periodica conversazione avente come denominatore comune quello della statistica, introduciamo uno degli aspetti fondamentali di questa scienza: la possibilità che essa ci offre di prendere decisioni su una base logica, non quella logica, per intenderci, basata soltanto sul buon senso ma anche, e principalmente, su delle basi matematiche tese a valutare i fatti oggettivi.

Nella vita quotidiana tutti noi siamo spesso chiamati a prendere delle decisioni e, istintivamente, di solito, ognuno di noi di fronte a una decisione tende a fare delle ipotesi, volte ovviamente a stabilire un rischio, o un maggiore guadagno derivante dal verificarsi di una certa ipotesi e, conseguentemente, da una certa decisione.

Bene, uno degli aspetti più importanti della statistica è costituito dall'esame di accettabilità di una ipotesi.

Ogni decisione è basata su due fatti fondamentali: le informazioni disponibili e il rischio che personalmente si è disposti ad accettare nel caso la decisione presa si dimostra alla fine errata.

Concettualmente la cosa sembrerebbe semplice, in pratica ci saranno alcune cose su cui dovremo soffermarci per chiarire i concetti, aiutandoci magari con esempi chiarificatori.

DICHIARAZIONE DI UNA IPOTESI

In statistica vengono considerati due livelli di ipotesi.

L'ipotesi sulla H_0 costituisce l'ipotesi della mancanza di differenze tra le caratteristiche del campione e quelle della popolazione reale. La reale esistenza di questo effetto o differenza, in altre parole il disaccordo tra i dati, viene chiamata ipotesi di ricerca ed è denominata H_1%

A questo punto bisogna stabilire il livello di significatività da dare a un test statistico, perciò occorre prima definire il rischio, che chiameremo α di accettare la possibilità, o probabilità, di rifiutare l'ipotesi nulla H_0 nel caso in cui essa invece sia vera. Di solito viene fissato al 5% e questo significa che, a priori, riteniamo di prendere una decisione corretta con il 95% di probabilità di essere nel giusto.

LA DISTRIBUZIONE NORMALE

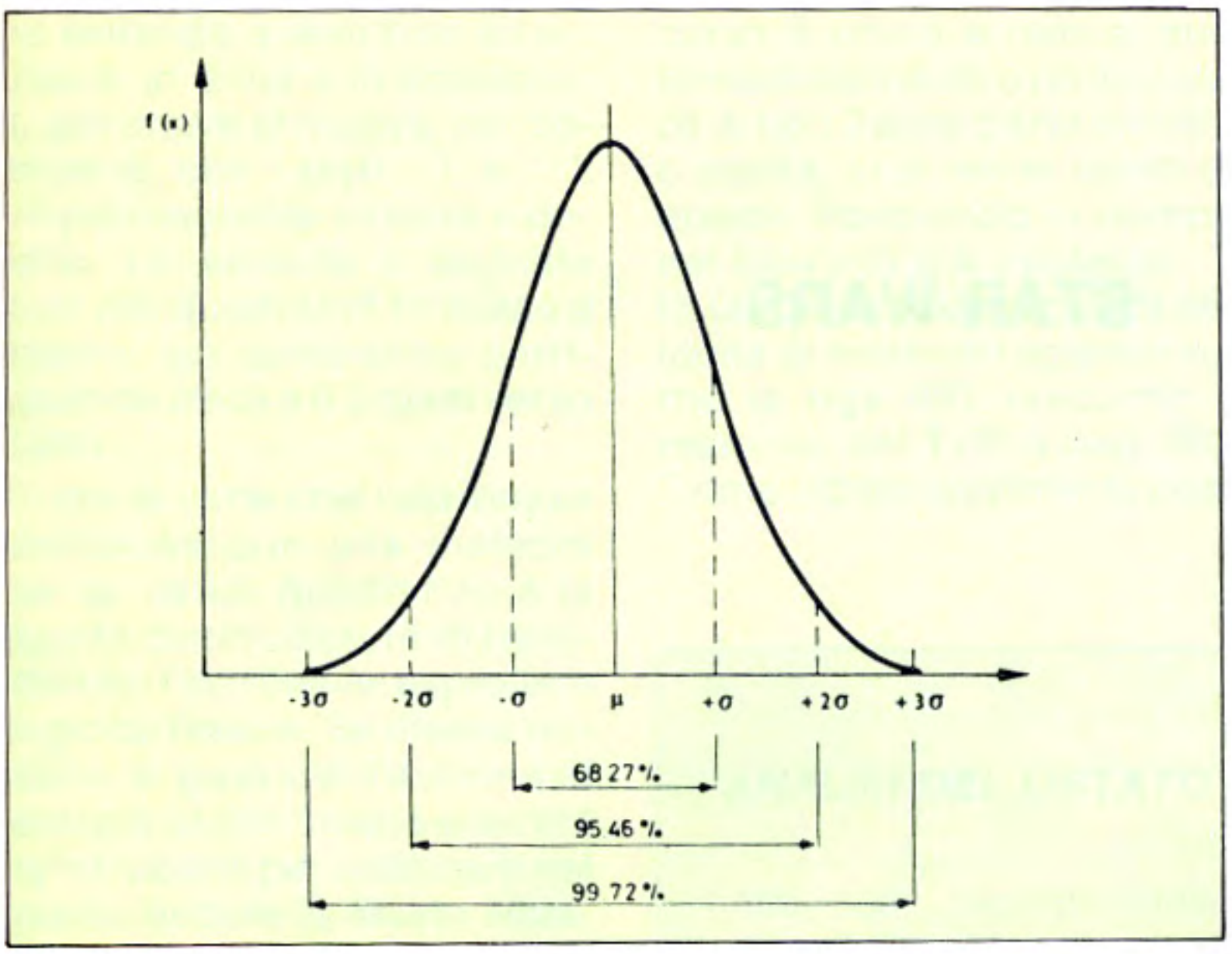
Quando si ha ragione di ritenere che le misurazioni di un certo fenomeno siano costituite in parti da cause costanti e in parte da errori di misurazione provocati da effetti casuali e non sistematici si può ragionevolmente ritenere che la distribuzione delle misure di quel fenomeno sia normale.

In tal caso la media delle misure tenderà, dopo un numero abbastanza ampio di repliche, ad avvicinarsi sempre di più alla media vera mentre un numero ristretto di esse si disporrà in funzione della deviazione standard.

In grafico sotto chiarirà il concetto di distribuzione normale e di deviazione standard delle misure.

Funzione di distribuzione di una serie di misure delle quali μ è il valore medio e σ la deviazione standard. È possibile vedere della figura il valore X compreso tra (x—3s e (x+3S).

A questo punto possiamo iniziare con il nostro:



CONFRONTO TRA DUE MEDIE DI GRANDI CAMPIONI

Presentiamo un programma che ci permetterà di comparare le medie di due grandi campioni: con questo programma ci sarà facile decidere, al livello del 95%, quale procedura sia più adeguata alle nostre esigenze, quale macchina fornisce le prestazioni migliori, e così via.
Vale la pena di otare che: 1° il numero di letture deve essere abbastanza grande (almeno 30) - 2° è necessario conoscere la derivazione standard di ciascun gruppo.
Per il resto è veramente facile impiegare il programma nel senso che, note due medie e m1 e m2 e le loro deviazioni standard s1 e s2, per decidere se m1 sia probabilmente migliore di m2, e viceversa, basta soltanto verificare la diversità tra loro; se il nostro calcolatore deciderà per la reale differenza tra le medie saremo autorizzati a credere che la media migliore sia davvero significativa.
Un appunto: siamo in grado di calcolare la deviazione standard poichè è stato recentemente pubblicato su questa rivista un programma per il calcolo di essa, ma se i campioni disponibili sono meno di 30?
Nessuno problema, introduciamo allora i:

TESTS F E T.

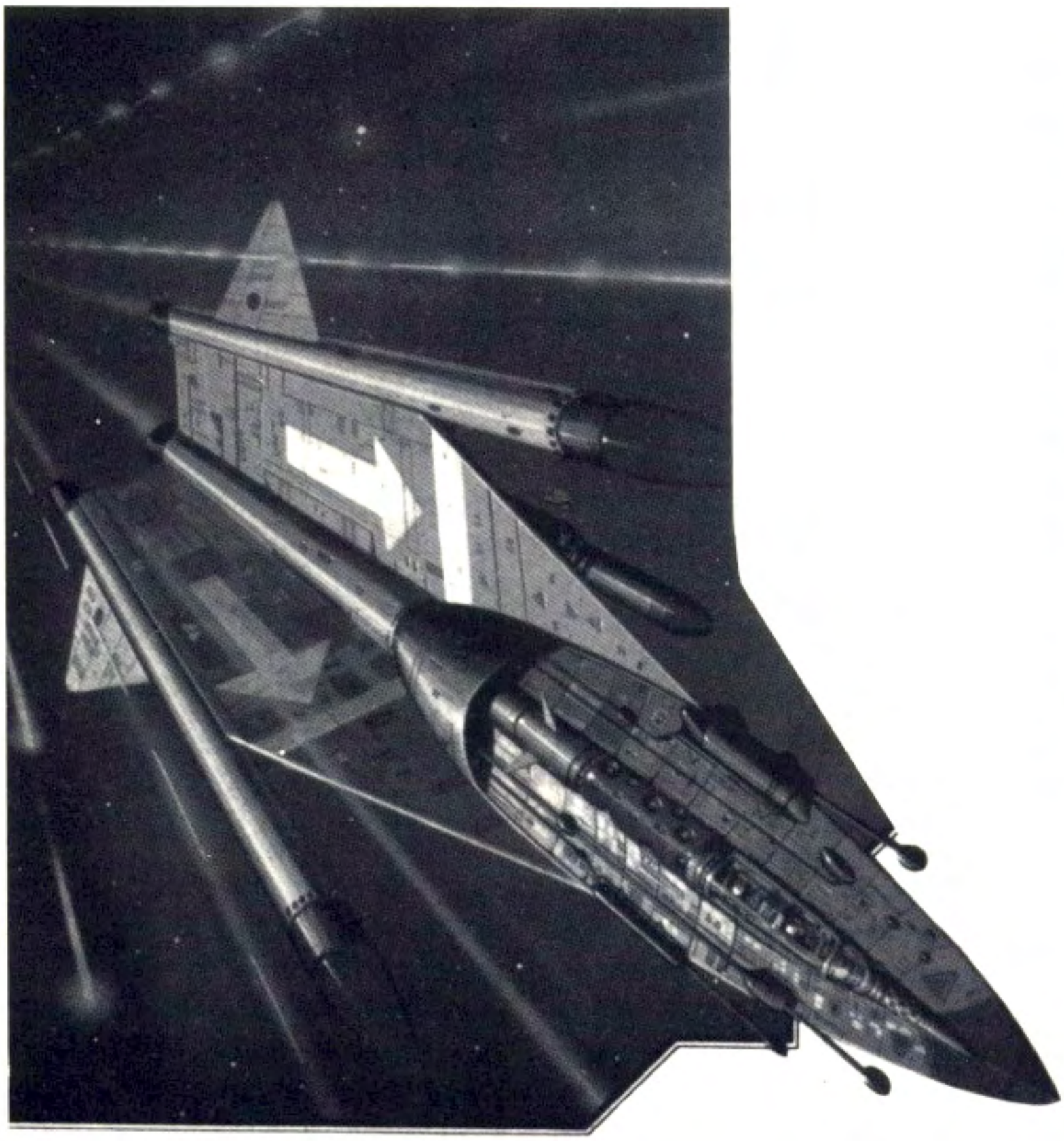
Il test T ci serve per paragonare delle medie costituite da piccoli campioni che presentino anche medie diverse ma di cui si presume che la deviazione standard, anche nel caso in cui essa non sia nota, sia uguale.
Ricorrimao ad un esempio per comprendere meglio l'uso effettivo dei tests F e T.
In un laboratorio è stata eseguita, da uno stesso analist a, la determinazione del Calcio presente nell'acqua di un fiume, sono state eseguite due serie di determinazioni su due campioni presi in momenti diversi e sono stati ottenuti i seguenti risultati.

Campione 1	Campione 2
21,8 mg/1	23,9 mg/1
24,5 mg/1	28,4 mg/1
23,3 mg/1	26,9 mg/1
27,6 mg/1	30,4 mg/1
23,2 mg/1	26,4 mg/1

La media dei risultati del campione 1 è uguale a 24,1, mentre per il campione 2 essa risulta 27,2.
Si tratta di stabilire se le concentrazioni risultano uguali o se esse siano effettivamente diverse.
Per fare ciò fissiamo anzitutto un livello di significativa del 5%, con il che avremo il 95% di probabilità di assumere una informazione corretta, e quindi ricorriamo al nostro programma.
L'analisi del test F ci dice che la probabilità che le varianze sinao tra loro correlate è molto alta, pari cioé al 42,7%; possiamo percio ritenere che le deviazioni standard siano uguali e che il test T sia significativo.
Il test T ci dà invece una probabilità pari al 6,5% riteniamo dunque che, con una possibilità di errore uguale ad un massimo del 5%, sia possibile accettare l'uguaglianza delle medie, in altre parole esse non differiscono in modo significativo e, pertanto le determianzioni sono tra loro coerenti, le concentrazioni di Calcio nel campione in esame sono effettivamente uguali.
Nel caso il test F ci dia una probabilità inferiore all'uno per cento avvertiamo che dobbiamo ritenere il test T non significativo, che le popolazioni pertanto abbiano una diversa deviazione standard inficiando le conclusioni ricavate dal test T.

PAOLO CIVARDI

STAR WARS



STAR WARS si compone principalmente di tre quadri. Il labirinto dentro la Morte Nera, la fascia dei meteoriti, e l'attacco finale.
Dopo la musica di presentazione basterà premere un qualsiasi tasto per incominciare.
Per muoversi nel labirinto saranno necessarie le frecce (sinistran destra, avanti). Esso si compone di trenta stanze in tutto e la strada per l'hangar è una sola. Vi sono delle stanze con il segno del "AT" sulla porta, ciò significa che avete sbagliato percorso.
Dopo essere passati attraverso la porta vi ritroverete di nuovo alla stanza di partenza; al terzo errore finisce il gioco.
La "P" su una porta indica che avete trovato le prigioni e quindi che il percorso sino a quel punto è esatto, mentre la H indica che siete arrivati all'hangar. Andando avanti si passa al quadro successivo.
La fascia di meteoriti si supe-

ra evitando questi fino ad arrivare in cima allo schermo. L'astronave si muove, per comodità, con i tasti "Q" e "/", rispettivamente sinistra e destra. La velocità è segnata con dei quadratini in basso a destra, un quadratino corrisponde circa a 0.5 pixel verso l'alto.

Tutte le volte che l'astronave collisiona con una meteora se ne va un quadratino e la spinta diminuisce, finiti i quadratini l'astronave esplode e il gioco finisce. Se invece riuscite a passare l'astronave entrerà in iperspazio e avrete le istruzioni per attaccare nel punto debole la Morte Nera. Il terzo quadro si divide in due parti. Nella prima bisogna arrivare alla fine del corridoio evitando e distruggendo i caccia nemici, nella seconda sparare per centrare lo sfogo radioattivo. Ci si muove su è giù con le frecce, per evitare i caccia nemici, e si spara con il tasto A; qui ogni collisione è fatale. Arrivati alla fine del corridoio bisognerà sparare, premendo space, quando i due mirini saranno sovrapposti.

Ogni volta che il gioco finisce lo schermo si cancella, premendo Y si ricomincia a giocare.

Il labirinto può essere modificato cambiando i data alle righe 320-330; dei sette numeri il primo è il tipo di stanza, il secondo e il terzo la stanza che si raggiunge andando diritti, il quarto e il quinto andando a sinistra e gli ultimi due andando a destra.

I tipi di stanze sono: 1 con porta, solo avanti; 2 a T, destra o sinistra; 3 a sinistra, solo sinistra; 4 a destra, solo destra.

Occorre ricordarsi che il percorso è unico e non si può tornare né indietro né in stanze di tipo 2 dalla parte sinistra o destra, ci si rende conto di questo tracciando la mappa del labirinto già esistente.

In ultimo si può cambiare velocità ai meteoriti agendo sul rnd di riga 690, (secondo il manuale del TI99 a pag. 98).

Come ultimo commento posso dire che la ripetizione della musica, ogni volta che si ricomincia il gioco, sebbene possa sembrare una scocciatura facilmente eliminabile, penso che debba essere come uno scotto da pagare.

RICCARDO GINEVRI

ANALISI DEL LISTATO

100- 130	Scritta e stelle
140- 240	Musica
250- 270	Arresto scritta e inizio gioco
280- 290	Recensione I° quadro
300- 310	Assegnazione colori e caratteri
320- 330	Data per il labirinto
340- 380	Data per disegnare le stanze
390- 410	Caricamento data
420- 440	Routine principale
450- 530	Routine per disegnare le stanze
540- 570	Recensione II° quadro
580- 630	Assegnazione caratteri
650	Data esplosione astronave
640	Data scritta iniziale
670- 740	Routine principale
750	Controllo collisioni e aggiornamento
760- 810	Esplosione e messaggio
820- 830	Iperspazio
840-1110	Assegnazione colori e caratteri III° quadro
1120	Data palazzi
1130-1150	Inizializzazione quadro
1170-1210	Routine principale
1220-1240	Routine sparo
1250	Distruzione astronave
1260-1270	Distruzione astronave nemica
1280	Assegnazione colori e caratteri fine corridoio
1290-1340	Disegno
1350-1360	Partenza mirini e rutine sparo
1370-1380	Controllo colpo
1390-1400	Sbagliato
1410-1470	Centrato e distruzione base nemica
1480-1490	Routine per rigiocare
280- 530	I° quadro
540- 830	II° quadro
940-1470	III° quadro

NOTE DI NATALE

Volete tanti regali da Babbo Natale?
Bene! Siete un aiutante di Babbo Natale e dovete guadagnare i regali che verranno distribuiti a Natale suonando le note intorno a voi, nella sequenza che compone un famoso motivo natalizio.
Dopo aver ascoltato l'intero motivo vi verranno riproposte le note: prima una, poi due, tre, e così via. Ogni volta dovete suonare le note che vi vengono proposte, se riuscirete a suonarle tutte guadagnerete tanti regali quante sono le note suonate. Se sbagliate dovete ricominciare tutto dall'inizio.
Per facilitare l'individuazione delle note, queste si coloreranno nel momento in cui si suonano.
Ora viene la parte più faticosa per voi: suonare.
Per realizzare questa operazione è necessario che il buon vecchio Babbo Natale, (è vero non gli assomiglia molto!) che inizialmente si trova nel centro dell'insieme di note, si sposti sulla nota che voi reputerete opportuno suonare; lo spostamento sarà possibile con i joysticks o con i tasti.
Quando Babbo Natale sarà esattamente posizionato sulla nota, potrete farla suonare premendo Enter, per chi esegue il gioco con i tasti, o il tasto rosso per chi lo esegue con i joysticks.
Buon Natale!!!

RICCARDO ROSSI

ADATTAMENTO AD UN POLINOMIO

Il concetto di funzione costituisce il fondamento di gran parte della matematica e della fisica.
Un'espressione come 5x+2 non ha una valore numerico definito finchè non si attribuisca un valore alla x. Si dice che il valore dell'espressione è dato in funzione della variabile x e si scrive f(x) = 5x+2. Ad ogni valore della variabile x corrisponde un ben preciso valore di f(x); ad esempio, per x=1 è f(x)=7, per x=2 è f(x)=12, ecc.
Quella appena considerata è una relazione di tipo matematico, tuttavia il concetto di funzione ha un campo di applicazione ben più vasto. 10000 lire del 1970 hanno lo stesso potere d'acquisto di lire 18000 del 1975 e di lire 50000 del 1980. I dati riportati consistono in tre coppie di valori e possiamo affermare che essi danno un certo valore monetario in funzione degli anni. Anche qui il termine funzione è usato in modo del tutto appropriato. Certo, a molti potrebbe interessare quanti soldi equivalgono attualmente alle 10000 lire del 1970! La risposta non può es-

DESCRIZIONE DEL LISTATO

100- 690: Titolo e definizione dei caratteri programmati per la grafica. Alla linea 210 vengono dimensionate alcune variabili con indice delle quali, in particolare, X() e Y() contengono le coppie di valori indicati dall'utente nella parte iniziale del programma.
700- 800: input del numero di coppie di valori e linee di controllo.
810- 950: ciclo per la scrittura dei dati iniziali.
960-1090: scelta del grado del polinomio.
1100-1340: formazione di una matrice con i coefficienti di un sistema di equazioni lineari.
1350-1730: soluzione di un sistema di equazioni lineari.
1740-1840: scrittura dei risultati consistenti nei coefficienti del polinomio.
1850-2070: calcolo e scrittura del grado di approssimazione raggiunto, espresso in valore percentuale.
2080-2230: menu per la scelta della continuazione.
2240-2450: input di un valore per la variabile x e calcolo del corrispondente valore della funzione f(x).
2460-2580: input degli estremi dell'intervallo di rappresentazione grafica.
2590-2750: routine per il dimensionamento automatico del grafico.
2760-2830: predisposizione dello schermo con la stampa degli assi cartesiani di riferimento e del massimo e minimo relativi della funzione nell'intervallo considerato.
2840-3010: ciclo per la stampa dei punti che formano il grafico della funzione.
3020-3120: istruzioni per creare un cursore lampeggiante che segnala il termine dell'esecuzione grafica.
3130-3170: subroutine per il calcolo della funzione in corrispondenza di un determinato valore di x.
3180-3240: subroutine per tracciare gli assi cartesiani, quando, ovviamente, cadono nella parte visibile dello schermo.
3250-3290: simulazione dell'istruzione PRINT AT.
3300-3310: fine del programma.

Avete acquistato il Numero di Dicembre

Ma sapete che sono usciti i Numeri di Giugno Settembre Ottobre e Novembre?

Per riceverli direttamente a casa vostra, scriveteci! Ve li invieremo in contrassegno a L. 10000 (+spese cadauno).

TI 99 NEWSOFT
Via S. Jacini, 4
20121 MILANO

serci fornita direttamente dai pochi dati a nostra disposizione. Se invece si conoscesse una relazione matematica che legasse le due variabili (lire e anni), il compito sarebbe relativamente facile.
Soprattutto se si dispone di un computer!
Effettivamente, date un certo numero di coppie di valori, è possibile ricavare una funzione matematica che si adatti ad esse con la maggiore approssimazione possibile.
Il programma «Adattamento ad un polinomio» assolve lo scopo di trovare la funzione polinominale, di grado n, che meglio si adatti ai valori in possesso dell'utente.
Una funzione polinominale è una funzione del tipo:

$$Y = a_0 x^0 + a_1 x^1 + \ldots + a_{n-1} x^{n-1} + a_n x^n$$

Il programma non prende quindi in considerazione altre funzioni, come quelle esponenziali, trigonometriche, logaritmiche, ecc., tuttavia consente di operare con un polinomio fino all'ottavo grado.
Il metodo seguito si basa sul calcolo dei valori che rendono minima la somma degli scarti quadratici e sulla soluzione di un sistema di equazioni lineari.
Dopo il titolo si deve specificare quante coppie di valori sono disponibili. Il programma accetta fino a 50 coppie ma, come è comprensibile, al loro aumentare si allunga il tempo necessario all'elaborazione. In seguito si dovranno scrivere tutte le coppie di valori avendo cura di separare con una virgola ogni elemento dal suo corrispondente.
Terminata la scrittura dei dati si specificherà il grado del polinomio usato per la loro interpolazione. Il grado deve essere minore del numero delle coppie e, come è già stato detto, non può essere superiore a 8. Chi usa il programma non deve temere di sbagliare perché esso è protetto contro eventuali errori per i quali invia opportuni messaggi.
Dopo un breve tempo per l'elaborazione appaiono i coefficienti del polinomio ordinato secondo le potenze crescenti della variabile x. Di seguito appare un valore percentuale che esprime la bontà dell'adattamento. Valori superiori al 90% si possono ritenere soddisfacenti, tuttavia, in alcuni casi, se è possibile, si potrà ripetere l'operazione con un polinomio di grado superiore.
Premendo un tasto qualsiasi si accede poi al menu per la scelta della continuazione. Il tasto 1 fa ripetere l'elaborazione con un polinomio di grado diverso (non è escluso che si possa usare lo stesso grado semplicemente per rivedere i coefficienti del polinomio). Il tasto 2 permette il calcolo della funzione per determinati valori della x. In tal modo si ottiene l'interpolazione dei dati o la loro estrapolazione a seconda che i valori della x cadano all'interno o all'esterno dell'intervallo dei valori iniziali.
Per uscire da questa sezione di lavoro si deve battere la lettera F ed il tasto ENTER in risposta alla richiesta di un valore per la x.
La terza scelta del menu prevede la stampa del grafico della funzione polinomale a media risoluzione e con il dimensionamento automatico. La media risoluzione si ottie-

PRINCIPALI VARIABILI

X(), Y():	contengono le coppie di dati iniziali.
P():	vettore usato per predisporre la matrice sulla quale si basa la soluzione del sistema di equazioni lineari.
F():	memorizza i valori della funzione in corrispondenza delle 64 colonne virtuali dello schermo.
TIME:	variabile di controllo usata nei cicli che creano delle pause.
ND:	numero totale delle coppie di dati in possesso dell'utente.
G:	grado del polinomio.
G2:	variabile di lavoro per la formazione della matrice.
N:	numero dei coefficienti del polinomio.
J, K:	variabili di controllo usate nei cicli.
R():	variabile con indice usata nella formazione della matrice.
A (J, K):	matrice bidimensionale contenente i coefficienti delle equazioni che formano il sistema.
V():	coefficienti del polinomio.
L, I, W, Q:	variabili di lavoro usate per memorizzare temporaneamente dei risultati.
T:	valore percentuale del grado di adattamento del polinomio ai dati iniziali.
XX$:	contiene il valore di x specificato dall'utente.
XV, YV:	valori numerici di x e corrispondenti valori di f(x).
FLG:	variabile flag che indica la possibilità di ricavare il valore numerico della stringa XX$.
X1, X2:	estremi dell'intervallo della rappresentazione grafica.
Y1, Y2:	minimo e massimo relativi della funzione nell'intervallo specificato.
DX, DY:	unità di misura relative ai due assi cartesiani.
RO, CO:	numero di riga e di colonna per la stampa degli assi ortogonali.
Z$:	variabile contenente la stringa da inviare alla subroutine PRINT AT.
RIGA:	numero di riga da trasmettere alla subroutine PRINT AT.
R1, R2:	numeri di riga per la stampa dei punti appartenenti alla stessa colonna.
DIFF:	variabile di lavoro per il calcolo del livello al quale si trova un punto all'interno di una riga.
P1, P2.	numeri compresi tra 1 e 4, individuano il carattere più opportuno, tra quelli definiti inizialmente, per rappresentare un punto del grafico della funzione.

ne con una routine originale che divide virtualmente lo schermo in 96 righe e 64 colonne. Rispetto ad una routine per la stampa in alta risoluzione (192x256 pixel), questa ha il vantaggio di essere estremamente più veloce. Infatti è necessario attendere un breve tempo per il dimensionamento automatico del grafico, in modo che esso non cada parzialmente fuori dello schermo, poi la stampa avviene nel giro di alcuni secondi.

Per il grafico della funzione è sufficiente specificare gli estremi dell'intervallo da rappresentare; il programma calcola il valore massimo e minimo della funzione e fa coincidere tali valori con i limiti superiore ed inferiore dello schermo. Questo comporta, in genere, due differenti unità di misura lungo i due assi cartesiani. Per avere qualche riferimento, oltre agli assi appaiono anche i valori massimo e minimo della funzione nell'intervallo specificato. Un cursore lampeggiante nell'angolo in basso a destra segnala che il grafico è terminato. Alla pressione di un tasto qualsiasi si ritorna al menu.

Per prendere familiarità con il programma si specifichi 4 come numero di coppie da inserire e successivamente si scrivano i seguenti valori: —2, —1 —1, 3 1, —1 2, 3. Si ponga uguale a 3 il grado del polinomio.

Dopo breve tempo appariranno i coefficienti 1, —3, 0, 1 dai quali si deduce che il polinomio interpolatore è:

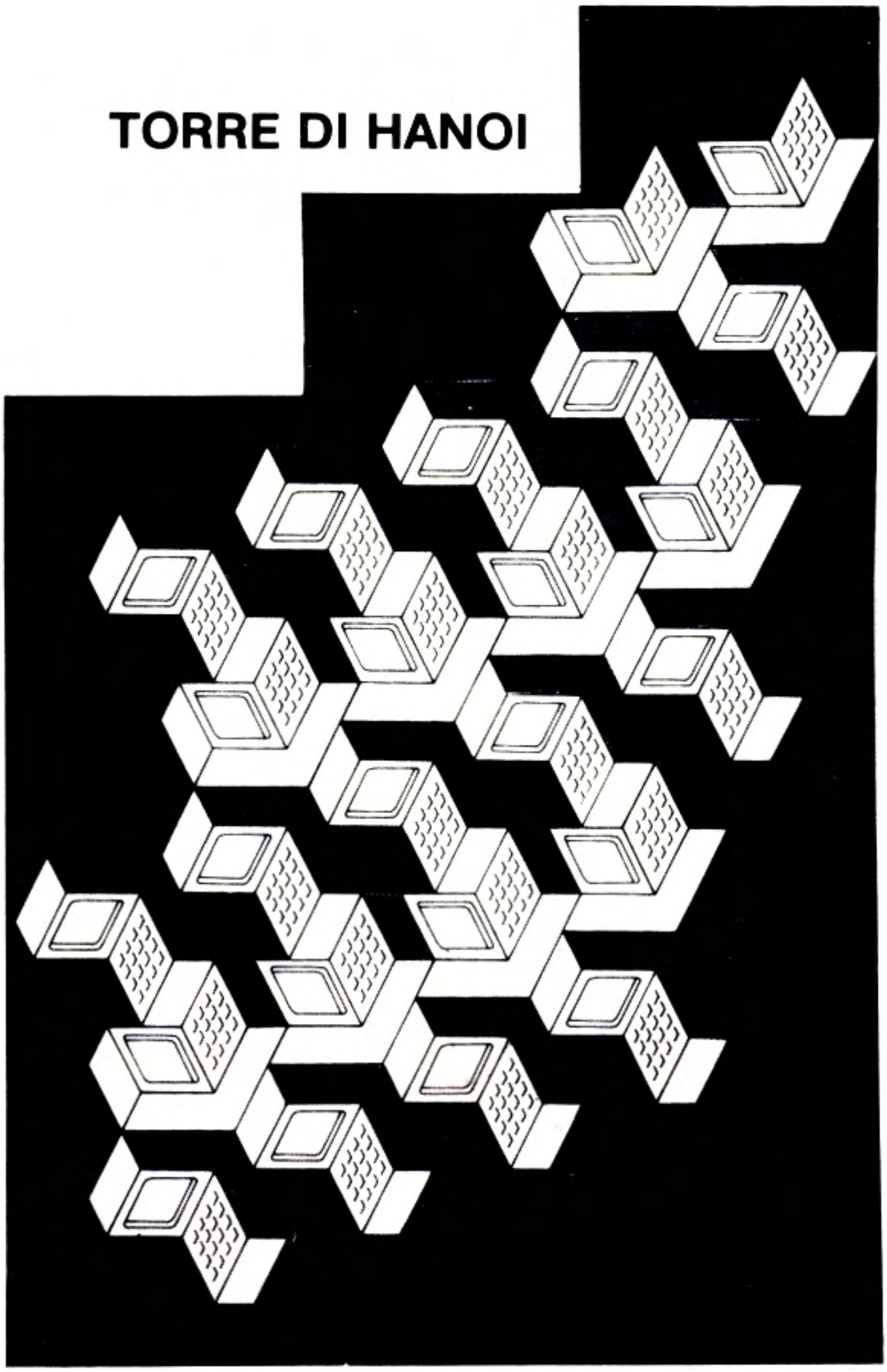
$x^3 - 3x + 1$

Battendo un tasto si passa al menu e scegliendo la continuazione 3 si passa al grafico. Indicati gli estremi —2 e 2 appare il messaggio UN MOMENTO, PREGO che precede il calcolo per il dimensionamento automatico.

Il programma traccia poi una curva passante per i punti aventi per coordinate le coppie di numeri scritte nella parte iniziale.

SERGIO BORSANI

TORRE DI HANOI



La torre di Hanoi è un edificio formato da vari piani multicolori, ognuno di larghezza decrescente proseguendo dal basso verso l'alto. Lo scopo del gioco è di ricostruire la piramide inizialmente nella prima posizione, in una delle altre posizioni 2 o 3 a scelta del giocatore. Nel ricostruire la torre, è possibile spostare il piano superiore da una posizione all'altra purché il piano spostato appoggi a terra o su un piano di larghezza maggiore. Negli altri casi lo spostamento non è consentito e si perde una mossa. La torre va ricostruita nel numero minore possibile di mosse. Per spostare un piano da una posizione all'altra basta battere prima la posizione di partenza (1,2 o 3) e poi la posizione di destinazione.
Inviate una foto dello schermo con i punteggi più bassi che riuscite ad ottenere. Ma attenzione! non alterate il programma per fargli mostrare un punteggio ottenuto... barando in BASIC.

GEY

DESCRIZIONE DEL PROGRAMMA

Linee	Descrizione della funzione
100-440	Generazione del quadro sullo schermo
450-564	Inserzione dati per lo spostamento del piano
570-670	Validazione mossa e spostamento del piano
680-760	Musica e finale al completamento della torre

NATALE IN "TI"

Il programmma in TI basic è molto semplice e cerca di sfruttare e valorizzare al massimo le capacità musicali del nostro TI 99/4A, il quale, pur non avendo la possibilità di variare il timbro, concede di poter usare tre voci contemporaneamente e separatamente in modo molto elastico per quanto riguarda, sia la dinamica (variazione di volume o intensità),sia la melodica (frequenza). È uno dei miei primi programmi, per cui mi perdonerete questa serie interminabilie di CALL SOUND del resto inevitabile volendo ottenere contemporaneamente all'esecuzione della melodia la visualizzazione graduale di una immagine. Lascio a voi il compito di riscriverlo e migliorarlo utilizzando le istruzioni FOR NEXT, READ e DATA, cosa che consentirebbe di variare la velocità di esecuzione anche da INPUT.
Il programma si articola in tre parti, nella prima viene eseguita STILLE NACHT a tre voci e nel contempo appare sullo schermo un albero realizzato in bassa risoluzione con i caratteri * e ∂ è una serie di PRINT interpolati tra le CALL SOUND e quindi appare la scritta BUON NATALE carattere per carattere realizzata con una serie di VCHAR, quindi lo schermo cambia di colore e l'albero e poi la scritta cominciano a lampeggiare, dulcis in fundo, TU SCENDI DALLE STELLE sempre a tre voci.
Se lo schermo nell'ultima parte risulta monotono, fate sbizzarrire la vostra fantasia interpolando tra le CALL SOUND di TU SCENDI DALLE STELLE delle HCHAR o VCHAR dopo aver definito i caratteri con delle CALL CHAR in modo da costruire un bel cielo stellato e una cometa od altro, ricordandovi di mettere alla linea 1710 un bel CALL CLEAR e di inserire una istruzione alla volta tra una CALL SOUND e l'altra in modo da non interferire nella continuità della musica.
Alla fine del programma, quando le ultime note si saranno spente e voi sarete lì commossi, vedrete l'albero che insiste a star lì sullo schermo, niente paura, il programma è fermo alla linea 3110 con un GOTO 3110. Premete, allora il tasto FCTN e contemporaneamente il tasto 4, otterrete un break quindi digitate 3110 GOTO 100, il programma, dopo il RUN, ricomincerà e continuerà a girare finché non ne potrete più.
È meglio ora che vi auguri BUON NATALE e buon ascolto ed a risentirci presto.

FERNANDO GIGLIOTTI

ANALISI DEL LISTATO

150 - 700	Prime note di STILLE NACHT, assegnazione dei colori e stampa dell'albero
710 - 810	Solo CHALL SOUND
820 - 1220	Assegnazione dei colori ai caratteri e loro stampa
1230 - 1570	Completamento del motivo
1580 - 1700	Lampeggio dell'albero e della scritta
1720 - 3100	TU SCENDI DALLE STELLE

Non esistono variabili, se non K e J dei due cicli FOR NEXT per ottenere il lampeggio.

DONI CELESTI

Il tipico scambio di doni e la tradizione dell'albero di Natale hanno radici antichissime. Nei tempi preistorici il solstizio d'inverno (che cade appunto nel periodo natalizio) era atteso non senza un superstizioso timore destinato poi a trasformarsi in sfrenata allegria. Allo scopo di aiutare il sole a ritrovare la sua lucentezza e forza, proprie del periodo estivo, si ricorreva a uno strano espediente: sulle cime delle alte conifere delle foreste del nord, che per ultime trattenevano i quasi spenti raggi del sole morente, si accendevano fascine resinose che illuminassero le notti del «terrore» e soprattutto volessero richiamare verso il settentrione, con la potenza della magia, l'astro benefico. Quanto a Babbo Natale (Santa Klaus nei paesi anglosassoni) si sa che deriva dall'italianissimo San Nicola, la cui pronuncia latina è Santus Nicolàus, e che aveva fame di essere generoso elargitore di doni per i bisognosi. Il programma, evidentemente ispirato a queste due tradizioni, riassume in sé quella che si può definire l'essenza del Natale. La presentazione del gioco con le note salienti per il suo utilizzo è accompagnata da una tipica aria natalizia che, vista la sua durata, è stata drasticamente soppressa nel caso

Lista variabili

A$	=	Definizione grafica del pacco
C$	=	Definizione grafica delle palline colorate dell'albero
B$	=	Stringa contenente i vari messaggi
CS	=	Variabile di flag per l'emissione della musica
K	=	Contenuto del valore ASCII del tasto premuto dalla tastiera
S	=	Status
CR	=	Colonna del pacco
RR	=	Riga del pacco
NR	=	Nuova posizione del pacco
P	=	Contenuto del numero dei pacchi
FL	=	Variabile di flag per la scelta del colore del pacco
N	=	Riga delle nuvole
D	=	Colonna del carattere di Babbo Natale
AA	=	Colonna del primo carattere delle nuvole in una FOR-NEXT
BB	=	Colonna del secondo carattere delle nuvole nella FOR-NEXT
X	=	Riga delle montagne
Y	=	Colonna delle montagne
AN	=	Colonna usata per l'animazione della slitta
ANB	=	Colonna usata per l'animazione di Babbo Natale
F	=	Status del joystick
CA	=	Carattere letto nelle GCHAR
SC	=	Carattere scelto per il pacco
Z	=	Carattere letto per verifica presa pacco
T	=	Variabile di tempo
FP	=	Variabile di flag per ripetizione del motivo sonoro
TE	=	Durata della nota
FR	=	Frequenza nella nota 1, 2,...
DU	=	Durata nella nota 1, 2,...
R$	=	Contenuto dei primi due caratteri letti nella stringa B$
C$	=	Contenuto del secondo e terzo carattere di B$

DONI CELESTI

Remarks

210 - 660	Definizione caratteri
670 - 950	Schermata di presentazione
960 - 990	CALL KEY e JOYST d'attesa
1000 - 1120	Definizione colori
1130 - 1230	Definizione variabili principali
1240 - 1300	Visualizzazione delle nuvole
1310 - 1640	Visualizzazione delle montagne
1650 - 1780	Movimento della slitta con Babbo Natale
1790 - 1850	Movimento di Babbo Natale quando scende dalla slitta
1860 - 1950	CALL KEY e JOYST per il controllo di Babbo Natale
1960 - 2020	Spostamento a destra
2030 - 2070	Spostamento a sinistra
2080 - 2160	Caduta del dono natalizio
2170 - 2460	Accorgimenti grafici e incremento di alcune variabili
2470 - 2540	Scritte in seguito alla raccolta dei pacchi
2550 - 2620	Ridefinizione dei colori
2630 - 2730	Visualizzazione dell'albero di Natale con le palline colorate
2740 - 2770	Richiesta di altra partita
2780 - 2840	Fine partita
2850 - 2990	Discesa delle nuvole in caso di mancata presa
3000 - 3070	Scelta casuale del codice carattere del pacco
3080 - 3430	Definizione del motivo musicale
3440 - 3540	Subroutine di stampa messaggi

voleste rigiocare. Premuto un qualsiasi tasto o il pulsante di fuoco dei controlli a distanza ecco avanzare su uno sfondo di montagne la veloce slitta di Babbo Natale. Dopo una rapida corsa al centro dello schermo il simpatico vegliardo, comandato con i tasti S e D o con i joystick (I), si adopererà per catturare quanti più pacchi possibili che, è proprio il caso di dirlo, piovono dal cielo. In caso di mancata presa il dono in questione provocherà, oltre all'abbassamento delle nuvole, uno smottamento del terreno che bisogna avere l'accortezza di sanare o con la barraspazio o con il fuoco dei joystick poiché se si verificasse che un secondo pacco cadesse sulla traccia del primo la partita avrebbe termine. Per consolarvi apparirà un verde e rigoglioso abete formato famiglia illuminato da palline colorate lampeggianti ed entrerà in funzione la nota musichetta. Una volta presa coscienza del punteggio conseguito nel prendere i doni vi verrà data l'opportunità di ricominciare la partita. Nel caso che la vostra scelta cadesse su no, riceverete i miei auguri personali (meglio che niente...)

Buon Natale!

MARCO SQUINTANI

CAMPAGNA ABBONAMENTI

Abbonarsi Conviene!

Fino a DICEMBRE, l'abbonamento a 11 numeri di TI 99 NEWSOFT costa ancora L. 80.000

Affrettatevi, dunque! Non mancate questa grossa occasione.

Abbonarsi è facile!

Basta inviare un assegno o vaglia postale intestato a "NEWSOFT snc", Via Stefano Jacini, 4 - 20121 MILANO.

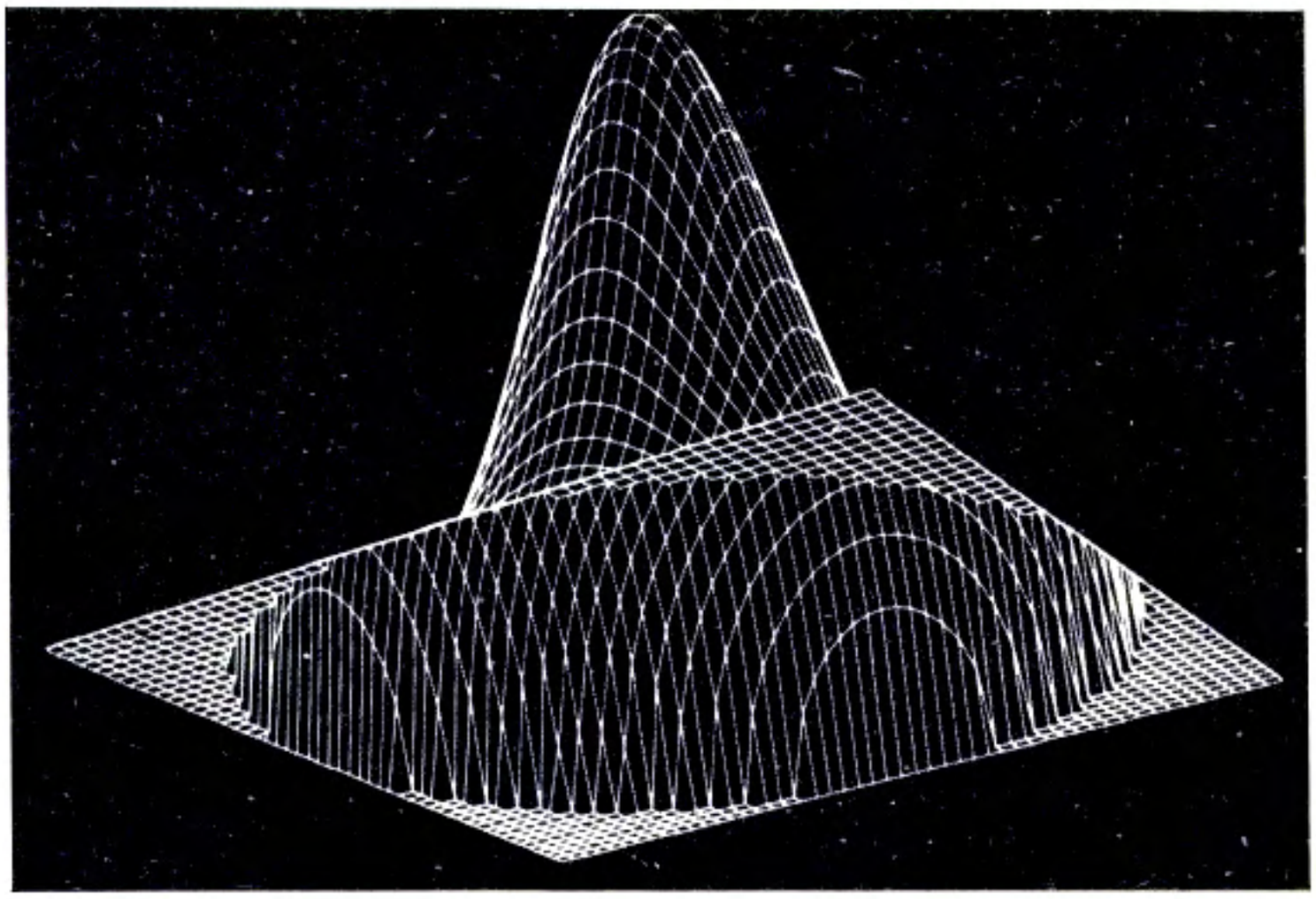
Dimenticavamo!

Gli abbonati potranno ottenere sconti e agevolazioni più favorevoli su alcune iniziative che presto vi "sorprenderanno".

SCROLL

Il nostro scopo era quello di creare un programma per gli sfondi in movimento da usare in basic, veloce e flessibile, che permettesse il movimento dello schermo in tutte le direzioni con o senza cambio di scena.
Il linguaggio usato è naturalmente l'assembler per mini memory (prossimamente verrà pubblicata anche la versione per extend basic). Il programma si carica con il programma L Easy Bug. La sintesi per chiamare il programma è la seguente: CALL LINK («SCROLL», ORIZZONTALE, VERTICALE, STRORR$, STRVER$), le parti racchiuse nelle parentesi quadre sono opzionali, in cui ORIZZONTALE e VERTICALE possono essere costanti, variabili o espressioni numeriche.
In ORIZZONTALE un valore negativo provoca uno scroll verso sinistra; zero non effettua scroll orizzontale; un valore positivo provoca uno scroll verso destra.
In VERTICALE un valore negativo provoca uno scroll verso il basso; zero non effettua scroll verticale; un valore positivo provoca uno scroll verso l'alto.
Le stringhe STRORR$ e STRVER$ sono opzionali essere omesse entrambe o solo la seconda. Queste stringhe servono per riempire le righe lasciate vuote dall'operazione di scroll. Se queste vengono omesse lo spazio viene riempito con la stringa di caratteri che esce dalla parte opposta dello schermo.
ESEMPI
CALL LINK («SCROLL», —1, —1, "XXX", "YYY")
in questo esempio viene effettuato uno scroll verso il basso e uno verso sinistra, nell'ultima colonna a destra, partendo dall'alto vengono stampate tre «X» e nella prima riga in alto vengono stampate tre «Y».
CALL LINK (SCROLL (1,—1)
in questo esempio viene effettuato uno scroll verso destra ed uno verso il basso, i caratteri che compongono la linea in basso vengono scritti nella linea in alto mentre i caratteri della colonna più a destra vengono riportati nella colonna più a sinistra.

RICCARDO ROSSI



MERGE

Come sapete il TI 99 in extend basic è in grado di effettuare il merge tra due programmi solo se ad esso è collegata una unità a dischi. I possessori di tale periferica, purtroppo, non sono molti; abbiamo perciò pensato di realizzare dei programmi che permettessero di fondere due programmi Basic con il solo registratore a cassette e una espansione di memoria che permettesse l'uso dell'assembler.
A tale scopo sono state scelte le seguenti configurazioni:

8MINI MEMORY
e
EXTENDED BASIC + ESPANSIONE 32K

Purtroppo per motivi di spazio siamo costretti a pubblicare in questo numero solo il secondo rimandandovi al prossimo numero di TI99 Newsoft per il merge per mini memory.

MERGE PER EXTENDED BASIC

Il programma per funzionare richiede, come già detto l'espansione di memoria da 32K (quella di TI99 market va benissimo se ancora non l'avete).

Se avete l'espansione collegata potete caricare il merge come un comunissimo programma basic. A questo punto dovete dare il RUN e il programma basic penserà a caricare l'assembler. Il merge si compone di due programmi richiamabili con i seguenti comandi:

CALL LINK («MERGE 1»)
e
CALL LINK («MERGE 2»)

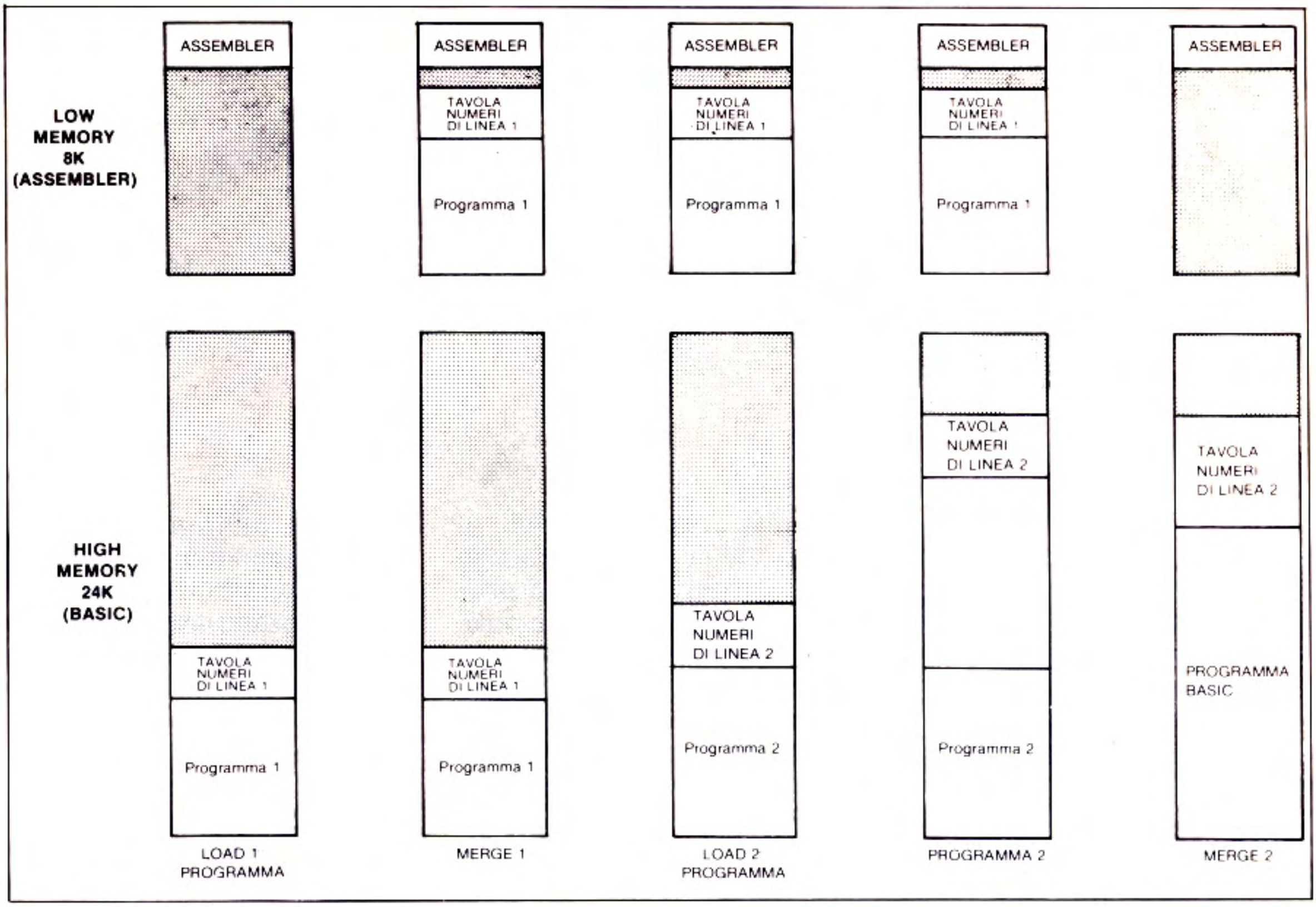
Prima di spiegare l'uso del merge vediamo di capire come funziona. Il programma utilizza quella parte di LOW MEMORY non utilizzata dal basic e non occupata dal programma assembler.come buffer per memorizzare temporaneamente il primo dei due programmi da mergiare (MERGE 1). La memoria disponibile è di circa 6,5K che sommati ai circa 13,5 caricabili normalmente dal registratore permettono di ottenere programmi della lunghezza di 20K. La fusione vera e propria dei programmi avviene con il merge 2. Con questo programma viene ripreso quello nella LOW MEMORY e unito al programma basic residente nella HIGH MEMORY. Per ulteriori chiarimenti guardate la figura 1.

Grazie alle decine di richieste che ci sono pervenute

ANNUNCIAMO che l'espansione di memoria da 32 K Byte costa ora L. 200.000 IVA compresa (180.000 per gli abbonati).

Chi l'ha richiesta al vecchio prezzo, la riceverà a casa pagando il nuovo prezzo.

Per ogni richiesta vedete a pagina 33.



COME UTILIZZARE IL MERGE

Abbiamo due programmi:

```
10 REM PRIMO PROGRAMMA
20 PRINT «PRIMO PROGRAMMA»
e
100 REM SECONDO
    PROGRAMMA
110 PRINT «SECONDO
    PROGRAMMA».
```

Ora li vogliamo unire con il MERGE.
Prima di tutto carichiamo il primo programma e battiamo CALL LINK («MERGE 1») in questo modo il primo programma viene copiato nella LOW MEMORY. Ora date NEW e caricate il secondo programma. Volendo ora noi fondere il programma in memoria con il precedente battiamo CALL LINK («MERGE 2»)
in questo modo viene eseguito il merge; se non ci credete provate a dare LIST il risultato sarà:

```
10 REM PRIMO PROGRAMMA
20 PRINT «PRIMO
   PROGRAMMA»)
100 REM SECONDO
    PROGRAMMA
110 PRINT «SECONDO
    PROGRAMMA»)
```

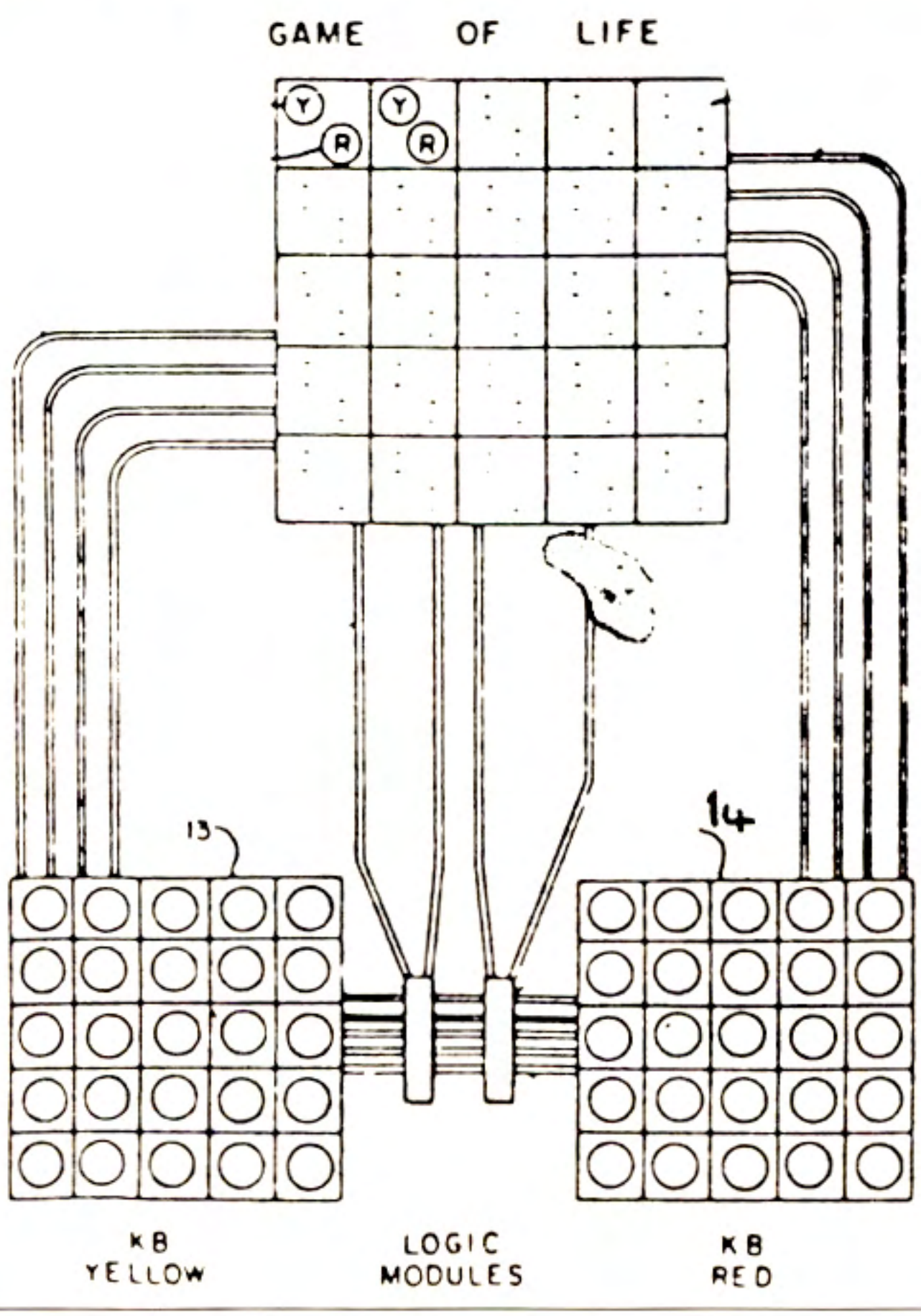
L'istruzione di MERGE 1, al contrario di MERGE 2, può essere data più volte di seguito per cui il programma memorizzato nella LOW MEMORY sarà quello dell'ultima CALL LINK. Questo vi permette, quando state battendo programmi molto lunghi, di riempire tutti e 6,5 K disponibili. Infatti quando il programma supera questa lunghezza al comando di MERGE 1 viene risposto *MEMORY FULL. A questo punto cancellate alcune linee e riprovate a dare il MERGE 1 fino a che tutto funziona regolarmente. Quando questo accade salvate il programma, date NEW e digitate la seconda parte del programma.
Eseguita anche quest'ultima operazione salvate il programma e date il MERGE 2.
Questo tipo di merge non gestisce numeri di linea uguali per cui il primo numero di linea del secondo programma deve essere più alto dell'ultimo numero di linea del primo programma contrariamente si possono verificare diversi inconvenienti.

RICCARDO ROSSI

GIOCO LOGICO DEL TIPO A TOCCO

BREVETTO ATARI INC. 019226

La presente invenzione si riferisce ad un dispositivo di gioco elettronico che utilizza un foglio di cartone in cui sono stati effettuati dei fori.
Una vernice conduttrice provvede ad entrambi i conduttori di interconnessione.
Detta vernice viene fatta fluire nei fori in modo che possano essere effettuati dei contatti a spina con le unità del microprocessore nei pacchetti di conduttori multi-spine e con gli indicatori a diodi emettitori di luce.
Inoltre sono predisposte le tastiere 13,14 mediante commutatori del tipo a tocco con una coppia di conduttori in cui il contatto è effettuato da vernice conduttrice su una membrana che viene abbassata dal dito.
Possono essere forniti indicatori fatti con materiale a base di cristalli liquidi in cui i colori sono fatti variare dal calore prodotto da una vernice «isolante» sul cartone collegato ad una coppia di conduttori verniciati.
Il gioco simula i cicli della vita.



LIBRI

35 PROGRAMMI PER IL TEXAS TI 99/4A - di Vince Apps - Franco Muzzio editore. Prezzo L. 10.000.

Ecco finalmente un libro sul TI 99 per gli amanti dell'EDIT and RUN. Si tratta infatti di una raccolta di programmi pronti all'uso, consigliati soprattutto a chi muove i primi passi nel mondo della programmazione. Quanto contenuto nel volume è di sicuro effetto e la semplice immediatezza dei programmi garantisce risultati soddisfacenti anche per i neofiti della programmazione. Le spiegazioni dei programmi sono chiare e sintetiche. Molteplici gli argomenti trattati, si va dalla classica battaglia spaziale fino ad arrivare ad un efficente data base domestico. Degno di nota il prezzo: meno di 300 lire a programma.

INCOMUNICABILITA'

Carissimi amici di TI 99 Newsoft, non immaginate nemmeno la mia felicità nel vedere finalmente in edicola una rivista interamente per il mio computer.
E' fatta davvero bene, complimenti.
Vi ho scritto però, soprattutto perché ho un grosso problema che non sono riuscito a risolvere, quello di caricare sul mio TI 99 i programmi della vostra cassetta.
Vi sarei grato se potreste aiutarmi.
Ciao, e alla prossima lettera!!!

MARCO BUCCI
Roma

Pubblichiamo la lettera di Marco Bucci di Roma per rispondere a tutti gli amici lettori che hanno avuto difficoltà nel caricare i programmi dalla cassetta allegata alla rivista. Molto spesso il problema è imputabile al cattivo funzionamento del registratore. Infatti il nostro TI 99 "non sopporta" variazioni di velocità nello scorrimento del nastro, o cattivi allineamenti della testina di lettura. Il problema del volume non crea particolari difficoltà. Quindi prestate attenzione ai messaggi di errore che compaiono sul video. Se ripetutamente, per più di un programma per facciata, il computer vi darà "ERROR - NO DATA FOUND", allora probabilmente la cassetta è difettosa (ne stampiamo migliaia, e può capitare una piccola serie difettosa). In questo caso vi preghiamo di volercelo restituire per posta solo la cassetta, provvederemo a sostituirvela.
Nel caso, invece, che il computer vi indica "ERROR DETECTED IN DATA", la cassetta funziona bene ed è il registratore a cassette ad avere problemi.
Comunque torneremo sull'argomento quando parleremo del "Registratore a cassette" nelle pagine della rivista.

PER ASPERA... OVVERO IL LINGUAGGIO DEL TI

Cari amici di TI 99 New Soft, innanzitutto complimenti per la Vostra super rivista (finalmente anche una per il TI 99/4A) molto interessante...
Vorrei fare una domanda: si può programmare sul TI 99/4A in linguaggio macchina? - Se sì, esiste un modulo, una cassetta o altro? Grazie.
Vorrei darvi un consiglio: perché non dedicate più pagine alla corrispondenza?

ANDREA LOMBARDO
Milano

Caro amico,
se hai seguito la rivista oramai avrai capito che è possibile programmare il TI 99 in linguaggio macchina.
Le possibilità sono tre: Modulo Editor Assembler + Espansione 32 K + disk drive; modulo Mini memory; Extended Basic + Espansione 32K.
Anche qui, tratteremo gli argomenti diffusamente nei prossimi numeri della rivista.
Per adesso, in risposta all'altra domanda, eccoti accontentato: da questo numero ben due pagine per rispondere alle vostre lettere.

PERCHE' VIVE IL TI 99...

Spett.le TI 99 Newsoft
Ho trovato qualche giorno fa in edicola la vostra rivista con cassetta e sono stato molto sorpreso ed entusiasta per aver visto questa "rinascita" del TI 99.

Ho posseduto vari "personal e home" computers, a partire dal "terribile" Sinclair ZX 81 al commodore CBM 64, completo di drive e stampante, ma l'unico che ho mantenuto e sul quale lavoro piacevolmente è l'Olivetti M 10.

Dalle esperienze di vari apparecchi posso dire che il TI 99, nei confronti di computers (spesso anche più costosi) ha un grande pregio: la definizione sul monitor, inoltre ha una semplicità e sicurezza nell'uso che è veramente unica!

Devo aggiungere che credo non si sia mai verificato che, dopo la cessazione di produzione di un oggetto o di un apparecchio, come in questo caso per il TI 99, vi sia stata una "rivoluzione" silenziosa

e spontanea da parte di coloro che, comprendendo i pregi e le caratteristiche eccezionali del TI 99, hanno continuato ad usarlo e hanno dichiarato apertamente di non condividere la decisione della casa produttrice, anzi, con la loro fiducia e costanza, hanno indotto altre case fabbricanti materiale elettronico, a produrre per il TI 99.
Ho espresso il mio entusiasmo, comunicato il mio grazie e, nella speranza di trovare sempre più spesso in edicola la vostra rivista, vi invio i più vivi complimenti e auguri!

GIORGIO MASSAZZA
Genova

Grazie dei complimenti!

...AD ASTRA PER I SUPER ESPERTI

Spettabile Redazione....
Vorrei pregarvi di trattare nei prossimi numeri l'argomento riguardante il richiamo da Basic di un programma assembly residente in Mini Memory. E' questo un problema per me ancora irrisolto, quando il programma assembly è preposto alla scrittura sul video di un testo tramite le direttive BLWP ∂ > 6028.

In tal caso infatti, mentre riesco in assembly a far girare il programma (tramite il RUN della Mini Memory), non riesco a visualizzare il testo da Basic tramite l'istruzione CALL LINK. Ho naturalmente caricato nella REF/DEF il nome del programma e il suo indirizzo di origine.
Il programmino è il seguente:

```
LI R 0 , 392
LI R 1 , ST
LI R 2 , 11
BLWP ∂ > 6028
B * 11
ST TEXT I HELLO AMICI'
END
```

STEFANO CONTESSA
Terni

Carissimo Stefano,
risolviamo il tuo problema: programmando in linguaggio macchina, volendo scrivere sullo schermo, bisogna ricordarsi che i caratteri hanno un codice diverso nei due ambienti (Basic e mini memory). Facendo partire un programma con il RUN della Mini memory, essi hanno il solito codice (in esadecimale); chiamando un programma da Basic, con il CALL LINK, bisogna addizionare > 60 o 96 al codice usuale (non è possibile perciò utilizzare la direttiva TEXT). Comunque, in un prossimo numero tratteremo esaurientemente l'argomento.

LA REDAZIONE DI TI 99 NEWSOFT AUGURA A TUTTI I LETTORI BUON NATALE E FELICE ANNO NUOVO

TI 99 MARKET

Siamo finalmente in grado di offrirvi i seguenti materiali. Tutti i prezzi sono comprensivi di IVA ed escludono i costi di spedizione e imballo contrassegno.

DESCRIZIONE	PREZZO	PER GLI ABBONATI
HARDWARE		
Home Computer TI 99/4A - 16bit - 16 KB completo di alimentatore e modulatore per collegamento TV con 2 moduli SSS gioco/educativi	229.000	200.000
Cavetto di collegamento al registratore	12.000	10.000
Cavetto adattatore Joysticks di qualsiasi tipo (Spectravideo...)	29.000	25.000
Joystick singolo	22.000	19.000
Interfaccia parallela esterna per il collegamento di stampanti	200.000	180.000
Cavo di collegamento interfaccia parallela esterna con qualsiasi stampante tipo Centronics mt. 1	36.000	33.000
Stampante Seikosha GP500A matrice 5x7, 50 crt/sec 80 colonne, normale ed espanso, grafica	549.000	499.000
Stampante EPSON RX80, 100 crt/sec grafica, 80/132 colonne bid. ott.	759.000	699.000
Stampante EPSON RX80 F/T come sopra gestisce anche foglio singolo	869.000	790.000
Monitor colori Hantarex CT900/2 con audio, 15" e cavo collegamento	589.000	530.000
Espansione di memoria esterna (non richiede il BOX) di 32 Kb ram	200.000	180.000
SOFTWARE		
TI LOGO in Italiano - Mondadori	179.000	162.000
Modulo Atari PAC MAN	69.000	60.000
Modulo Atari DEFENDER	69.000	60.000
Modulo EXTENDED BASIC	180.000	160.000
Modulo TERMINAL EMULATOR	49.000	42.000
Modulo PRG gest. dati personali	49.000	42.000
Modulo STATISTIC (tedesco)	19.000	16.000
Modulo ADD-SUB A/1 - educativo	19.000	16.000
Modulo MATH GAMES 6 - educativo	19.000	16.000
Modulo NUMERATION 1 - educativo	19.000	16.000
Modulo DEMOL. DIVISION - educativo	19.000	16.000
Modulo MATH GAMES II - educativo	19.000	16.000

segue

DESCRIZIONE	PREZZO	PER GLI ABBONATI
segue SOFTWARE		
Modulo EARLY READING - educativo	19.000	16.000
Modulo WUMPUS - gioco	19.000	16.000
Cassetta VIRGIN ROBOPODS	16.000	13.000
Cassetta VIRGIN FUN PAK	16.000	13.000
Cassetta VIRGIN FUN PAK 2	16.000	13.000
Cassetta VIRGIN FUN PAK 3	16.000	13.000
Cassetta BLUEGRASS corsa ippica	10.000	8.500
Cassetta DECATHLON	12.000	10.000
Cassetta GOBLINS REVENGE labirinto	10.000	8.500
Cassetta SNOW TREK corsa di slitte	10.000	8.500
Cassetta UP PERISCOPE batt navale	10.000	8.500
Cassetta FAMILY GAME 4 giochi	10.000	8.500
Cassetta OLDIES BUT GOODIES GAME	10.000	8.500
Cassetta PERSONAL FINANCIAL AIDS	10.000	8.500
OFFERTE KIT		
TI LOGO Mondadori in Italiano e espansione di memoria esterna da 32 Kb con libro: Impariamo a programmare in TI Basic	369.000	330.000
4 moduli da lit. 19.000 assortiti	59.000	54.000
9 cassette assortite	99.000	90.000
Stampante SEIKOSHA GP500A completa di cavo ed interfaccia parallela	699.000	660.000

Per Natale regalate un TI 99

TI 99 NEWSOFT

GENNAIO 1985 L. 10.000

EDISOFT

COSTRUIAMO UN ARCHIVIO SCACCHISTICO

COSA FARE DI 32 K IN PIU'

ASSEMBLER: Il Multicolor

IN SOFTWARE & SOFTWARE:
SCACCHI - Prova di memoria
Multicolor - Poker - Data Base

SOMMARIO

□ **EXTENDED**
Come fare per abbreviare il tempo di prescan nel TI99 — pag. 3-4

□ **HARDWARE**
A chi serve e come usare la memoria da 32 K esterna — pag. 6-7

□ **ASSEMBLER e L/M**
Il Multicolor — pag. 6-7

□ **TI 99/4A**
Cominciamo in questo numero a parlare della CPU — pag. 7-8

□ **TEACH BIT**
Un bel programma che integra l'articolo dedicato alla CPU. Gira in Basic — pag. 8

□ **MINI MEMORY**
Qui parliamo della ROM di cui è fornito il modulo MINI MEMORY — pag. 9-10

□ **MULTICOLOR**
Un programma per la MINI MEMORY con il quale potrete realizzare lavori grafici di qualsiasi tipo
A pagina 13 viene pubblicato il listato per i possessori del modulo EDITOR ASSEMBLER — pag. 11-13

□ **ARCHIVIO SCACCHISTICO**
Questo programma in Extended vi mette in grado di creare un completo archivio scacchistico.
Scritto da Sergio Borsani, prevede anche l'uso dei dischi e della stampante — pag. 14-16

□ **TI BASIC**
Continuiamo qui la trattazione dell'argomento "Gestione files su cassetta" — pag. 17

□ **BATTAGLIA NAVALE**
Giocate contro il computer una battaglia fino all'ultima nave — pag. 18

□ **POKER**
State attenti che può bluffare anche lui — pag. 19

□ **DATA BASE**
Createvi un archivio su misura con questo potente Data Base. Gira in Extended — pag. 20-22

□ **PROVA DI MEMORIA**
Sembra facile....! — pag. 23

□ **THE CAVE**
Fate si che il minatore perfori i due muri per passare allo schermo successivo — pag. 24

□ **STATISTICA**
Nel nostro viaggio in questo mondo di numeri, questa volta parliamo del coefficiente di correlazione R — pag. 25-26

□ **LA POSTA** — pag. 27-28

□ **PICCOLI ANNUNCI** — pag. 29-30

□ **COLLABORATE CON NOI** — pag. 31-32

□ **TI 99 MARKET** — pag. 33-34

TI 99 NEWSOFT
ANNO 2 - Numero cinque
GENNAIO 1985
MENSILE
Direttore responsabile:
ANTONIO C. LOSITO
Coordinatore Editoriale:
VINICIO CIVININI
Direttore Amministrativo:
ENRICO CALLERIO
Art Director:
BEPPE RE-FRASCHINI
Illustrazioni:
MARIA TERESA COCOZZA
Segreteria:
EMANUELA CHIAPELLO
CRISTINA FRISON
Redazione:
FIORENZA AURIEMMA
DARIO BAREZZI
Collaboratori:
SERGIO BORSANI,
GIANPAOLO BOTTIN,
DANIELE CATALFAMO,
PAOLO CIVARDI,
STEFANO DELLI PONTI,
FRANCESCO MACALI,
MAURO MATTIAZZI,
EZIO MONTINI,
LEVIO PEROTTI,
RICCARDO ROSSI,
MARCO SQUINTANI,
ALBERTO STRAFILE,
SEBASTIANO TOMASELLO
TI 99 NEWSOFT è una pubblicazione della **NEWSOFT snc**, via Stefano Jacini 4 - 20121 MI
Tel. 02/807464 - Telex 324284
REGISTRAZIONE AL TRIBUNALE DI MILANO N. 323 DEL 7.7.84

COPYRIGHT NEWSOFT snc
TUTTI I DIRITTI RISERVATI
Manoscritti e foto originali, anche se non pubblicati, non si restituiscono ed è vietata la riproduzione anche parziale di testi, fotografie e listati.

Pubblicità: AV STUDIO
Via Giuseppe Meda, 23
Tel. 02/8357155
Abbonamenti: l'abbonamento annuo a 11 numeri di *TI 99 NEWSOFT* costa L. 95.000 per l'Italia; L. 160.000 per l'Europa (spedizione aerea); L. 200.000 per tutti gli altri paesi.
L'importo, per assegno non trasferibile o per vaglia postale, va inviato a NEWSOFT snc, via S. Jacini 4 - 20121 Milano.
Stampa: CGB, Albegno di Treviolo (BG)
Concessionaria per la distribuzione:
EUROSTAMPA Srl - C.so Vittorio Emanuele 111 - 10123 Torino - Tel. 011/538166-7

L'editore non si assume responsabilità alcuna sul contenuto degli articoli e/o software inviati.
Per gli articoli firmati e/o siglati da collaboratori esterni, la redazione si assume la responsabilità prevista dalle leggi sulla stampa.

Tutti possono collaborare a TI 99 NEWSOFT, sia con articoli che con programmi. Scriveteci o telefonateci in redazione per avere tutti i ragguagli.

PRESCAN DEI PROGRAMMI

Ogni volta che vogliamo far eseguire un programma al nostro computer, dobbiamo attendere alcuni secondi per permettere alla macchina di eseguire uno scanner dei programmi.
La macchina compie questo scanner per trovare eventuali errori presenti nel programma, in caso la ricerca risulti negativa, egli precompilerà il programma, in caso contrario segnalerà l'errore.
In EX.BASIC, esiste un modo per eliminare tale spreco di tempo, dato che il computer compie sempre tale operazione, anche se il programma non è errato.
Il tempo di attesa, dato dallo scanner, varia notevolmente con il variare della lunghezza del programma; parte da un minimo di un secondo, per programmi di poche righe, ad un massimo di 40-90 secondi per programmi di grosse dimensioni. Le parole chiave !∂P-, !∂P+, permettono di disattivare e attivare lo scanner. Per poter utilizzare questa opzione, dobbiamo rispettare alcune regole:
a - dobbiamo elencare tutte le variabili e le CALL utilizzate nel programma prima di disattivare lo scanner (eventualmente, se usato, includere nella lista la istruzione OPTION BASE, specificando il valore 1 o 0).
b - dobbiamo assicurarci che il programma non contenga alcun tipo di errore, e che tutte le CALL e le variabili utilizzate siano state elencate;
c - le variabili e le CALL devono essere posizionate su una o più righe, spaziate dal caratteristico segno : : ;
d - il programma non deve passare per le righe dove è presente l'elenco, quindi sarà opportuno inserire un salto incondizionato «GOTO», alla prima istruzione eseguibile del programma, il GOTO può riferirsi anche alla linea dove è posizionato il segno !∂P-; dato che questo segno si trova alla linea prima della prima istruzione eseguibile.
e - elencando le CALL, non è necessario mettere gli attributi:
es. CALL HCHAR (10, 10, 34), potrà essere scritto: CALL HCHAR.
Detta regola vale PER TUTTE LE CALL.
f - Per quanto riguarda le variabili, dovremo assegnargli un valore:
es. A, B, C, D, FF, GR, ET=O, oppure se non desideriamo esplicitare il valore (dichiarare il valore), potremo scrivere: A, B, C, D, FF, ET=KF, utilizzando un'altra variabile non ancora elencata.
g - Eventuali DEF devono essere poste in linee singole, vedi esempio A, prima dell'esclusione dello scanner.
Queste regole valgono per la generalità dei programmi, ci sono poi dei casi atipici, cioè non previsti da tali regole.
Uno tra questi è l'uso delle (CALL nome del sottoprogramma).
Dette CALL .... devono essere elencate nella lista, mentre le SUB ... a cui si riferiscono non devono figurarvi.
Quando vogliamo chiamare una di queste routine, dobbiamo ripristinare lo scanner per permettere al computer di poterla trovare ed eseguire.
Il segno !∂P+, riabilita lo scanner.
Esso deve essere posto prima che la CALL venga eseguita, e prima della SUB interessata.

Dopo che la sub desiderata ha eseguito la sua funzione, potremo riescludere lo scanner con il segno !∂P-.

ESEMPIO A

Prescanner eliminato in programmi senza CALL e SUB.

```
10 GOTO 100 :: CALL CLEAR ::
A$=B$ :: A,B=C
90 !@P-
100 CALL CLEAR
110 PRINT "CIRCONFERENZA DI
UN CERCHIO": : : : : : : :
120 INPUT "DIMMI IL RAGGIO "
:B
130 PRINT "LA CIRCONFERENZA
E' :";(B*B)*PI
140 INPUT "ANCORA SI-NO ":A$
150 IF A$="SI" THEN 100
160 END
```

ESEMPIO B

Prescanner eliminato in programmi con CALL e SUB.

```
10 GOTO 100 :: CALL CLEAR ::
CALL HCHAR :: A$=B$ :: A,B=C
:: CALL SCHERMO ::CALL SCREE
N
90 !@P-
100 CALL CLEAR
110 PRINT "CIRCONFERENZA DI
UN CERCHIO": : : : : : : :
120 !@P+
130 CALL SCHERMO
140 !@P-
150 INPUT "DIMMI IL RAGGIO "
:B
160 PRINT "LA CIRCONFERENZA
E' :";(B*B)*PI
170 INPUT "ANCORA SI-NO ":A$
180 IF A$="SI" THEN 100
190 END
200 !@P+
210 SUB SCHERMO
220 RANDOMIZE
230 C=INT(RND*13)+3
240 CALL SCREEN(C)
250 SUBEND
```

- Figura 1 -

GLI ESEMPI DI FIGURA 1

Questi due esempi, rappresentano una delle condizioni tipiche in cui ci potremmo imbattere per attuare il procedimento sopra accuratamente descritto.
Essi avranno anche la funzione di chiarirci meglio le idee.
L'esempio A ci fa notare come deve essere correttamente usato il comando per la eliminazione dello scanner, e come devono essere elencate le variabili e le CALL.
Questo esempio serve per tutti quei programmi dove non c'è alcun sottoprogramma chiamato con CALL ....
L'esempio B, molto simile ad A, rappresenta l'esatto posizionamento dei segni !∂P- e !∂P+ in programmi dove sono presenti dei sottoprogrammi con BUS ... .

EZIO MONTINI

ESPANSIONE 32KBYTES

Finalmente, grazie a TI 99 MARKET, anche i possessori della configurazione minima del TI 99/4a, possono utilizzare l'espansione da 32 Kram.
Questa espansione viene connessa nella porta laterale del computer, e **non richiede l'utilizzo del costoso BOX di espansione.**
Per poterla utilizzare, dobbiamo essere in possesso di uno di questi moduli:

— EXTENDED BASIC
— MINI MEMORY
— ED/ASSEMBLER

Se alcuni di voi possiedono già il BOX di espansione, ma non sono riusciti a reperire l'espansione da inserire all'interno, possono tranquillamente connettere questa alla porta di espansione laterale, quindi inserire il connettore del BOX a lato dell'espansione da 32K.
Essa non crea alcun problema di compatibilità e permette l'utilizzo di tutte le schede o le periferiche collegate al BOX (unità a dischi, stampante, ecc..).

EXTENDED + 32K

La configurazione ideale per l'utilizzo dell'espansione è senza alcun dubbio quella composta da: TI 99/4a, EX-.BASIC
Essa ci permette di programmare in LINGUAGGIO MACCHINA, e di poter gestire un numero di dati ben superiore

a quello precedente.
La RAM è mappata dall'indirizzo >AOOO (-24576) a>FFFF (-1), in questa area di memoria, vengono immagazzinati i nostri programmi.
In realtà la memoria disponibile in BASIC, non è 32K, bensì di 24488 BYTS. Mentre in linguaggio macchina sono interamente disponibili.
Forse vi domanderete perché il conto dei K. disponibili da utente non torni: infatti se noi sommiamo 16K (presenti nel computer) + 32K (0 24... se si lavora in BASIC)=48K.
Quando inseriamo l'espansione, il computer riorganizza l'intera mappa di memoria, riservando 16K per le variabili e le matrici, e lasciando disponibile l'intera espansione, a cui dobbiamo togliere circa 7K richiesti dal computer per poter utilizzare il modulo EXTENDED.
Non avete sognato, prima ho proprio detto che è possibile programmare in linguaggio macchina, infatti ci è consentito l'utilizzo di tre nuove istruzioni:

— CALL INIT
— CALL LOAD
— CALL LINK

Queste tre istruzioni erano presenti anche prima, ma non venivano interpretate, generando di conseguenza dei segnali di errore.

CALL INIT

Questa istruzione opera un reset dell'espansione, cancellando un eventuale programma in linguaggio macchinam presente in memoria, essa non cancella i programmi basic.
Per poter utilizzare le istruzioni CALL LOAD e CALL LINK è necessario effettuare il reset almeno una volta dal momento dell'accensione.
Può essere utilizzata in modo comando o in un programma.

CALL LOAD (indirizzo, val. 1, val.2,..., val. n," ", indirizzo, val. 1, ..., val. n)

nota: val=valore da inserire all'indirizzo.

ovviamente il valore 2 sarà inserito automaticamente alla locazione successiva.

Questa istruzione svolge una funzione di scrittura nella CPU RAM, cioè nella memoria del microprocessore TMS 9900.

Essa può essere paragonata alla "POKE" degli altri computer, anche se molto più potente.

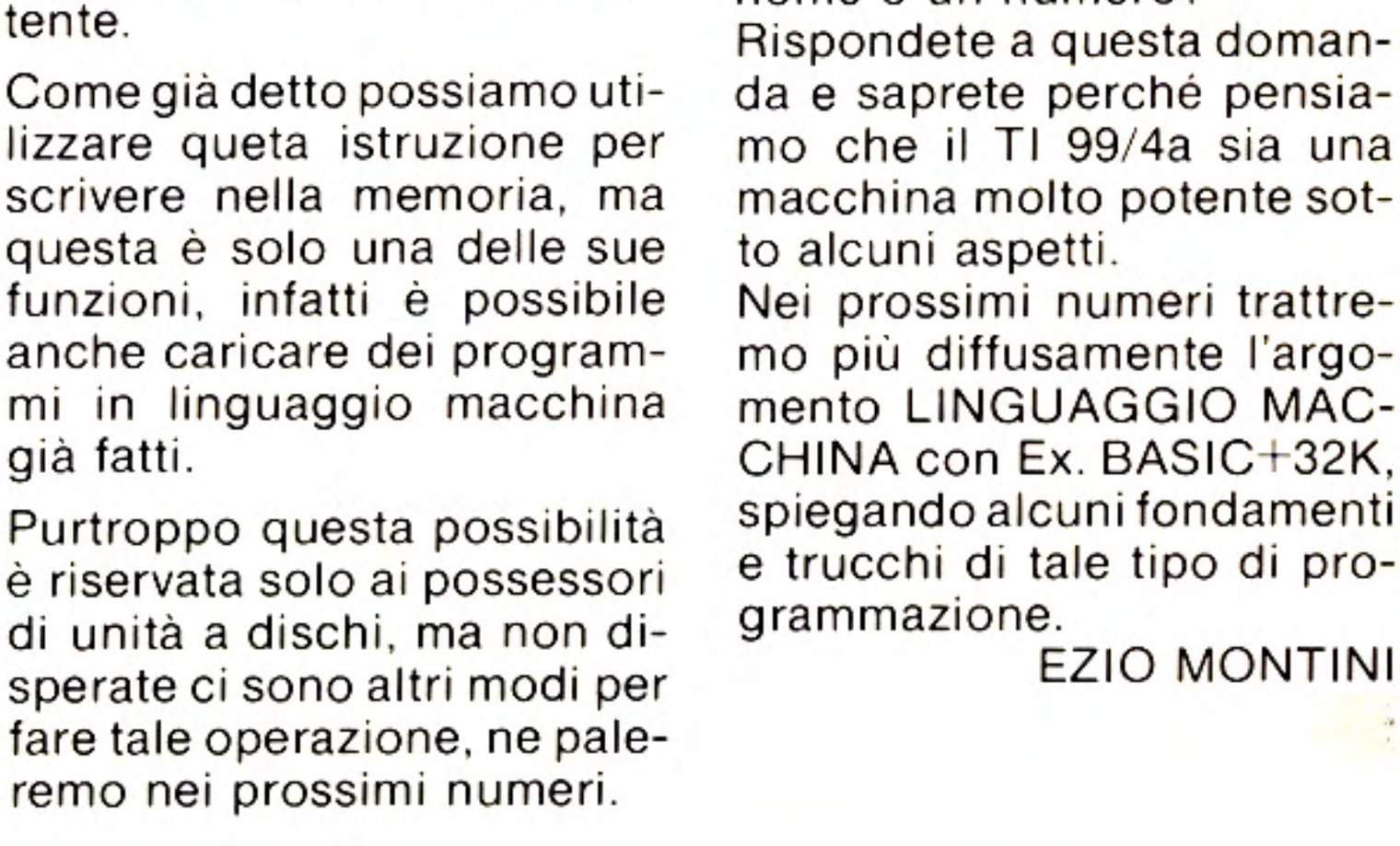
Come già detto possiamo utilizzare queta istruzione per scrivere nella memoria, ma questa è solo una delle sue funzioni, infatti è possibile anche caricare dei programmi in linguaggio macchina già fatti.

Purtroppo questa possibilità è riservata solo ai possessori di unità a dischi, ma non disperate ci sono altri modi per fare tale operazione, ne paleremo nei prossimi numeri.

CALL LINK ("nome del programma in L.M.")

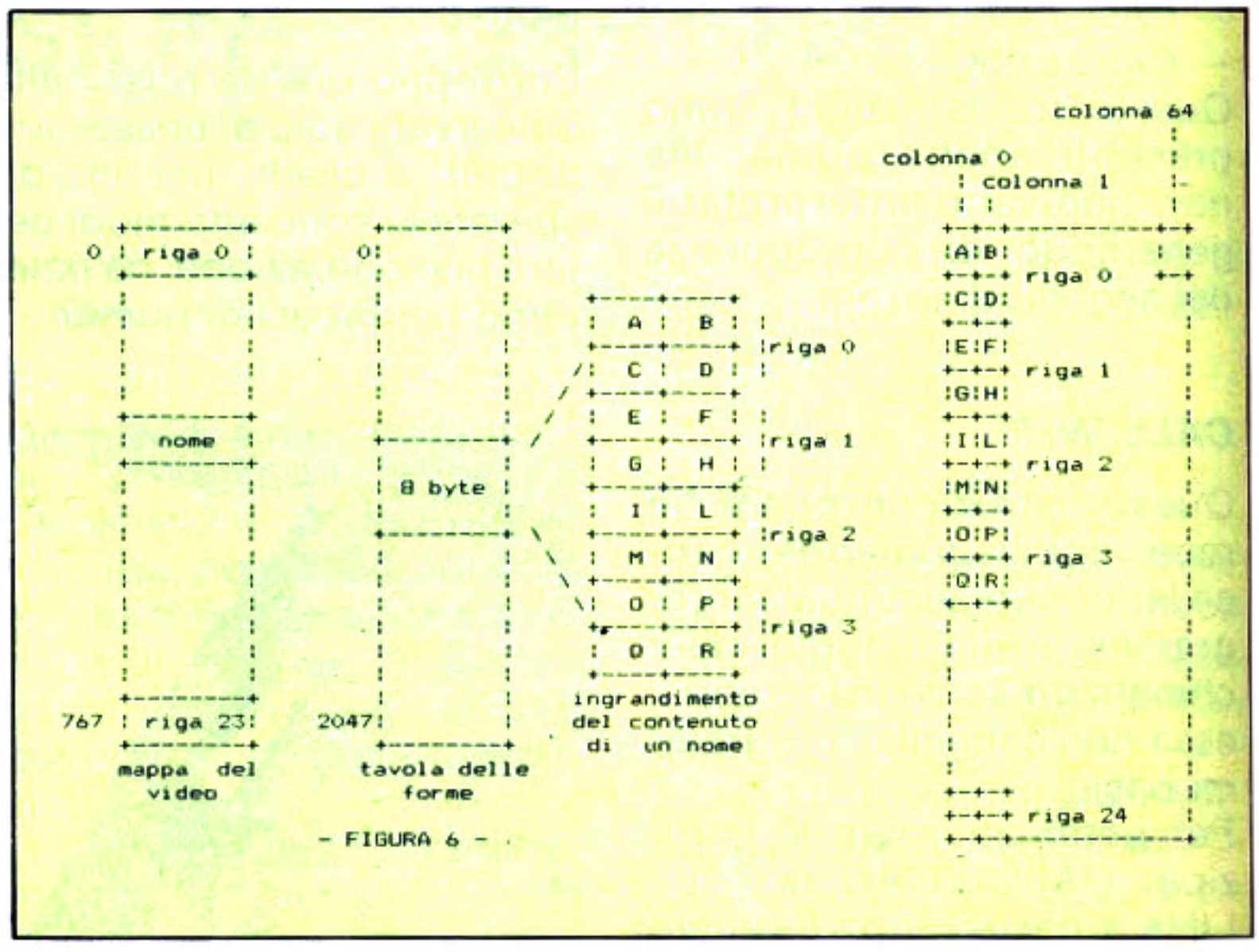
Non poteva certamente mancare un istruzione per fare partire il programma in L.M., in essa dobbiamo specificare il nome del programma in L.M. che vogliamo mandare in esecuzione.
Sempre per fare dei confronti, possiamo paragonare questa istruzione alla SYS di altri elaboratori, così facendo, non possiamo far altro che elogiare ancora una volta il nostro TI 99/4a, per la sua comodità di programmazione e potenza.
Infatti nei computer che fanno uso dell'istruzione SYS si deve specificare lo indirizzo di partenza del programma in L.M. .
Secondo voi è più semplice da mandare a memoria un nome o un numero?
Rispondete a questa domanda e saprete perché pensiamo che il TI 99/4a sia una macchina molto potente sotto alcuni aspetti.
Nei prossimi numeri trattremo più diffusamente l'argomento LINGUAGGIO MACCHINA con Ex. BASIC+32K, spiegando alcuni fondamenti e trucchi di tale tipo di programmazione.

EZIO MONTINI



MULTICOLOR

Ed eccoci giunti all'ultimo modo disponibile con il TMS 9929, il multicolor. Questa configurazione, è sostanzialmente diversa dalle altre, lo schermo video diminuisce la sua risoluzione che diventa di 48 per 64 quadratini colore.
Ciascun quadratino colore occupa lo spazio di 4x4 pixel, e il suo colore può essere uno qualsiasi tra i 16 disponibili.
La mappa dei nomi in multicolor, è identica a quelle dei modi grafici, e consiste in 768 byte, ognuno dei quali contiene il nome relativo a quattro quadratini colore, e fa riferimento alla tavola delle forme.
La tavola delle forme, in questo caso non si può certo dire che adempia alle sue funzioni, in quanto anzichè definire le forme, definisce i colori dei «quadratini», ma questo viene fatto in un modo per così dire «bizzarro». Infatti degli otto byte disponibili, solo due sono utilizzati per il colore del nome corrispondente. Certamente il discorso non è tanto semplice. Supponiamo di voler definire i primi quattro quadratini in alto a sinistra, quindi 2 nella prima riga e 2 nella seconda riga. Per prima cosa dobbiamo leggere il contenuto del primo byte della mappa dei nomi.
Supponiamo ancora che questo byte contenga «00», che quindi faccia riferimento ai primi otto byte della tavola delle forme. Orbene di questi otto byte, dovremo esaminarne solo i primi due, che descrivono rispettivamente il primo i colori dei due quadratini della prima riga, il secondo quelli della seconda riga. Ma non è finita qui, anche gli altri byte vengono utilizzati, ed ora, vedremo come. In multicolor, due gruppi di quattro quadratini, possono avere colori diversi, pur avendo lo stesso nome, e questo si spiega con il fatto che in multicolor il nome è legato alla riga video in cui deve essere visualizzato.
Riprendendo l'esempio precedente, supponiamo di avere lo stesso nome dei primi



- FIGURA 6 -

quattro quadratini («00») anche per altri quattro quadratini posizionati sotto i precedenti, quindi 2 quadratini nella terza riga, e due nella quarta riga.
Ebbene questi quadratini, avendo lo stesso nome dei precedenti, faranno ancora riferimento ai primi otto byte della tavola delle forme, ma questa volta, assumeranno i colori specificati dal terzo e quarto byte.
Così dicendo resta evidente che se lo stesso nome dovesse essere rappresentato nella quinta e sesta riga, i riferimenti saranno nel quinto e sesto byte, come saranno nel settimo e nell'ottavo, se il nome sarà nella settima e ottava riga.
Quindi lo schermo, è diviso in sei gruppi di quattro righe, e per ognuno, valgono le regole che abbiamo visto per il primo.
Come ogni volta, vi consigliamo di guardare bene la mappa (figura 6), in quanto vi aiuterà a comprendere meglio quanto detto.
E con questo, generalmente si dice, è tutto. Abbiamo visto tutte le configurazioni disponibili con il TMS 9929, ma ci siamo limitati a descriverne il funzionamento, e non siamo ancora in grado di utilizzarle.
Dal prossimo numero cominceremo a vedere qualche esempio in pratica, ed analizzeremo il significato dei registri del «VDP».

SEBASTIANO TOMASELLO

HARDWARE
LA CPU

Come il termine stesso suggerisce, hardware è tutto ciò che in un sistema di elaborazione possiamo toccare con mano. Lo è, per esempio, il computer stesso con i suoi componenti ma anche il video, la stampante, le memorie di massa (registratore e floppy disk) ecc. ecc. Per esaminare tutte queste parti è stato necessario operare un'ulteriore suddivisione. Considereremo quindi, di volta in volta, l'unità centrale di elaborazione, le memorie di lavoro e di massa, i vari dispositivi di entrata-uscita e qualche periferica.

LA CPU

La CPU acronimo per Central Processor Unit, o microprocessore, più che il cuore del calcoratore ne rappresenta il cervello. Frutto delle più avanzate tecniche di LSI, la nostra CPU, il **TMS 9900**, lavora ed elabora un numero impressionante di dati in razioni di secondo.
Vediamo ora in dettaglio il suo funzionamento: le istruzioni per effettuare i calcoli sono immagazzinate nell'unità di memoria e scritte sotto forma di programma specifico per un certo tipo di elaborazione. Il primo passo che compie il microprocessore è quello di leggere, una alla volta, le istruzioni ed estrarre dalla memoria i numeri da elaborare. Questa serie di dati viene inviata all'unità aritmetica e logica (ALU). Questa unità, interna alla CPU, provvede a svolgere materialmente le operazioni aritmetiche e logiche come, per esempio, i confronti tra numeri (A è minore o maggiore di B ecc.).
Anche la velocità di calcolo di questo componente è strabiliante; basti pensare al fatto che è in grado di svolgere un milione circa di moltiplicazioni al secondo.
Dai suoi circuiti elettronici i dati passano quindi nell'unità di memoria in una determinata posizione, in modo che sia possibile il loro recupero. Al termine di tutte le operazioni il microprocessore invia ai dispositivi di output (uscita) il risultato finale.
Possiamo ora riassumere le funzioni principali governate dall'unità di controllo della CPU in tre fasi: estrazione di istruzioni e di numeri, elaborazione, immagazzinamento e invio delle operazioni. Tutte queste attività devono già preesitere in memoria prima di rendere operativa la CPU. Perciò la memoria del computer è organizzata in celle contrassegnate da indirizzi specifici che le caratterizzano.
Ciascuna cella o locazione può contenere una quantità

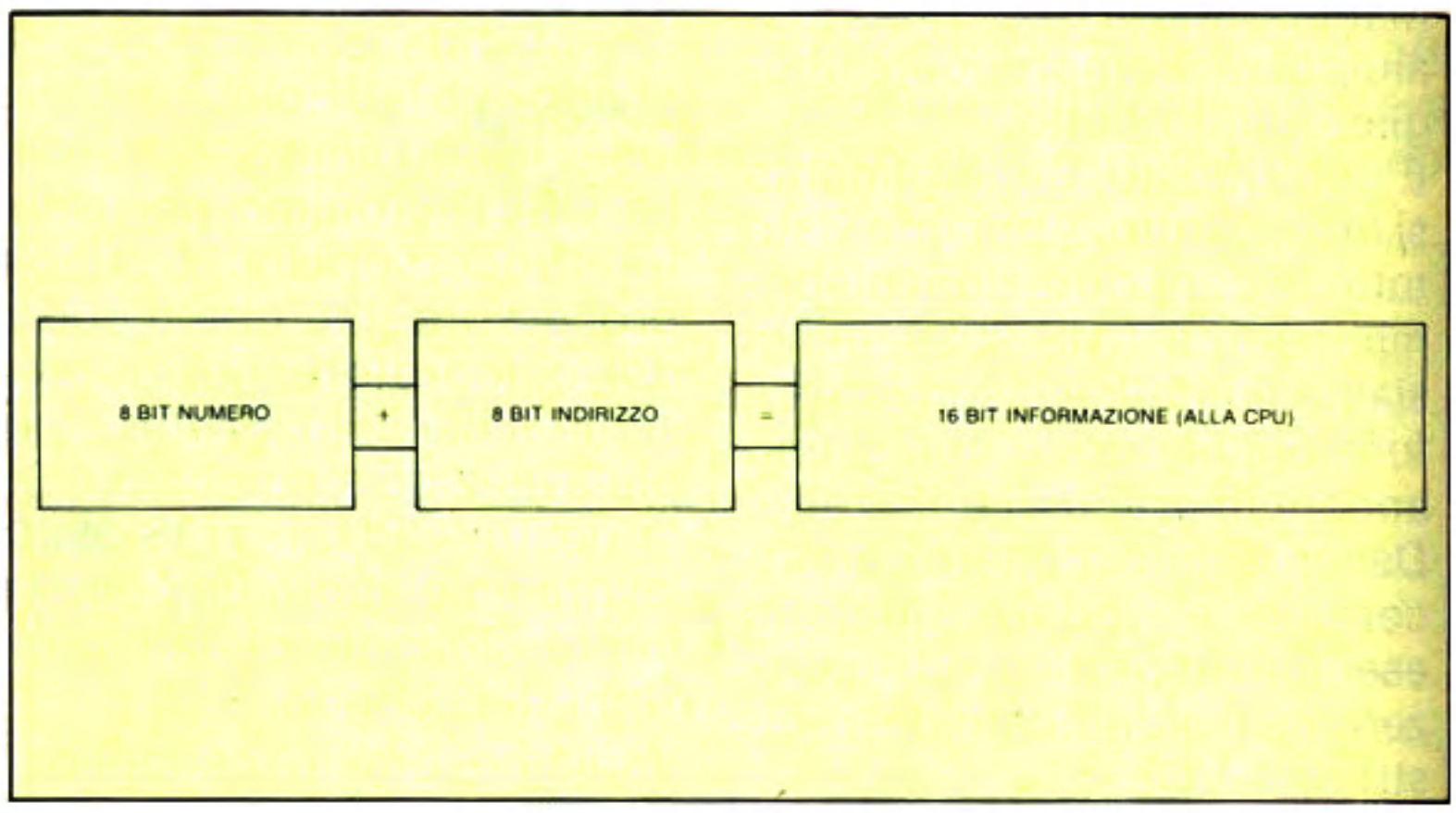
fissa di informazioni contrassegnate da un proprio indirizzo. Poiché il calcolatore è formato da un gran numero di circuiti elettrici, l'informazione dovrà essere data sotto forma di impulso elettrico. Ma come la macchina riconosce i vari impulsi? Grazie alla condizione di stato presente-assente (on-off) di questi impulsi, il computer attribuisce valore zero nel caso di assenza e valore uno nel caso contrario. È facile adesso creare una sequenza di cifre binarie (bit della contrazione inglese di binary digit), che può essere utilizzata per indicare lo stato degli interruttori del microprocessore.
La memoria del calcolatore è inoltre strutturata in modo che ogni locazione possa contenere una serie di otto bit (= 1 byte). Durante l'esecuzione di un programma di calcolo è tuttavia necessario fornire alla CPU l'indirizzo della locazione di memoria in cui quel valore deve essere immagazzinato; in caso contrario, dato il suo scarso potere decisionale, il computer si bloccherà inesorabilmente (figura 1). Questo numero, composto di altre otto cifre binarie, andrà ad affiacarsi al byte che contiene le cifre che devono essere elaborate.
Lo stesso discorso vale nel caso che il computer debba tradurre delle istruzioni che vengono introdotte e che compongono il programma.
Per sveltire tutta quest'opera di trasformazione binaria la CPU si avvale di apposite locazioni interne, dette registri.
L'ultimo problema che si pone è quello della sequenza delle istruzioni. Infatti, un'operazione non deve iniziare prima che si sia conclusa quella precedente. Questa procedura, detta appunto sequenziale, è assicurata dalla presenza di un altra inportantissima unità, posta all'interno del microprocessore: il clock, sul cui funzionamento ci soffermeremo nel prossimo numero. Intanto, memorizzate il tutto con il programma "teach bit" (e... non scandalizzatemi il Texas!).

MARCO SQUINTANI

TEACH BIT

Questo programma integra, in maniera perfetta, la rubrica «Hardware» e può costituire un ottimo modello per software redatto a fini educativi.
Dopo la presentazione su sfondo bianco-azzurro, partirà il programma vero e proprio. Alla sinistra del video si può contemplare (mi pare proprio un verbo azzeccato) un esempio di grafica veramente notevole che illustra l'interno del computer mettendo in luce il microprocessore e alcune ROM.
Il disegno è composto da ben 3136 pixel, suddivisi in 7x7 caratteri, un lavoro da certosino. Alla sua destra sono visualizzate le domande fatte, le risposte giunte e quelle sbagliate. Più in giù, subito dopo un asterisco, appare il quesito a cui bisogna rispondere e, ancora più in basso, le varie possibilità di risposta.
Per verificare il vostro grado di apprendimento, premete il tasto corrispondente a quella risposta che si ritiene essere esatta e quindi attendere. Nel caso abbiate risposto esattamente, comparirà una scritta di conferma e sentirete un suono acuto; in caso contrario, avrete un cupo suono di rimprovero e vi verrà svelata la risposta giusta.
Al termine delle dieci domande, avrete il resoconto della vostra situazione accompagnato dall'insindacabile giudizio del vostro Texas che non esiterà, nel caso dovesse cogliervi impreparati, a darvi una bella ... tirata d'orecchie.

M.S.



LA ROM

La Mini Memory, come sappiamo, è fornita di ROM e GROM, dove sono contenute numerose routines di cui l'utilizzatore può servirsi programmando in Basic o in LM. Nelle GROM sono memorizzati i codici necessari per implementare le nuove istruzioni Basic, quali LINK, PEEK e INIT; nella ROM troviamo le utilities che immancabilmente adoperiamo in Assembler (XSCAN, VSBW, VWTR, GPLLNK, LOADER, etc.).
In questo numero cominceremo ad occuparci della memoria a sola lettura nel modulo, **dei 4 K che vanno da >6000 a >6FFF.**

I VETTORI

I primi bytes della ROM (>6010->6052) sono occupati da vettori per il CONTEXT SWITCH, che operiamo con l'istruzione BLWP (o XOP, in altri casi).
Per capire quanto appena detto, bisogna ricordare cosa succede (a grandi linee, naturalmente) quando il microprocessore legge l'istruzione: facciamolo con un esempio.
BLWP @>6020. Sapendo cosa contengono le locazioni interessate (>6020 7092; >6022 6110), si ricava che la CPU sposterà l'area di lavoro a >7092 e comincerà ad eseguire le istruzioni da >6110 in poi.

Conoscendo i vettori (Fig. 1), si può quindi sapere dove sono allocate le diverse routines nella ROM e i loro Workspaces nella RAM.

Name	Workspace	Start Address
GPLLNK	>7092	>60F6
XMLLNK	>7092	>60C8
KSCAN	>7092	>6110
VSBW	>7092	>6126
VMBW	>7092	>6132
VSBR	>7092	>6140
VMBR	>7092	>614C
VWTR	>7092	>615A
DSRLNK	>7092	>61E4
LOADER	>70D8	>62EC
NUMASG	>70F8	>660E
NUMREF	>70F8	>66FE
STRASG	>70F8	>6768
STRREF	>70F8	>6888
ERR	>70F8	>6966
NAMLNK	—	>605A
TGOBLD	—	>62CA
CIF	—	>61BC

Fig. 1 - I Vettori.

LA NAME LINK ROUTINE

In >6010 scopriamo l'indirizzo di partenza della NAME LINK ROUTINE (>605A), della quale abbiamo già detto qualche cosa nel numero di Novembre. Ora, facendo riferimento al suo disassemblato (Fig. 2), ne spiegheremo più dettagliatamente il funzionamento. Essa, ripetiamo, si occupa di ricercare nella REF-/DEF TABLE il nome del programma chiamato (che viene memorizzato nell'accumulatore da>834A a >83FF). Passa in rassegna tutti i nomi a partire dall'indirizzo che trova in >701E; confronta a due a due i caratteri nella table con quelli nell'accumulatore. Se corrispondono, memorizza in >7020 l'indirizzo posto di seguito alle tre words destinate al nome; sposta l'area di lavoro a >7020 a RO ed, infine, salta al programma con BL * RO. Se, quando ha finito di scandagliare la tabella (R1 = 8000), non ha trovato il nome, ci segnala l'errore (>60AC— >OF è il codice di "PROGRAM NOT FOUND"). Se la lunghezza del program name è zero (se siamo all'opzione 2 del menù Mini Memory, ciò significa che abbiamo premuto l'enter senza aver digitato alcun carattere o soltanto spaces), la routine salta al programma che ha come start address il contenuto di >7020.
La subroutine in >65DA ha la funzione di controllare che tutte le references siano "resolved"(!).
Per questo numero ci fermiamo; nel prossimo continueremo il nostro discorso sulle routines in ROM, con lo scopo di raggiungere una discreta conoscenza del modulo.

Stefano Delli Ponti

Avete acquistato il Numero di Gennaio

Ma sapete che sono usciti i Numeri di Giugno Settembre Ottobre, Novembre e Dicembre?

Per riceverli direttamente a casa vostra, scriveteci! Ve li invieremo in contrassegno a L. 10000 (+spese cadauno).

TI 99 NEWSOFT
Via S. Jacini, 4
20121 MILANO

```
605A    MOV R11,    ∂>702E
605E    MOVB        ∂>8349, R1      salva il vecchio PC
6062    COC         ∂>6056, R1
6066    JEQ         >60BC
6068    MOV         ∂>8350, RO      muove in RO la lunghezza
                                    del program name
606C    JEQ         >6090
606E    BL          ∂>65DA          salta alla subroutine che
                                    verifiche se ci sono unresolved
                                    references
6072    JMP         >6080
6074    CI          R1,>8000
6078    JEQ         >60AC
607A    MOV         R1, R0
607C    LI          R2,>834A
6080    C           *RO+, *R2+      confronto name
6082    JNE         >60A6
6084    C           *RO+, *R2+
6086    JNE         >60A6
6088    C           *RO+, *R2+
608A    JNE         >60A6
608C    MOV         *RO, ∂>7020     salva in 7020 il nuovo Default
                                    Entry Address
6090    LWPI        >70B8           USRWSP
6094    MOV         ∂>7020, R0
6098    JEQ         >60AC
609A    BL          *R0             salta al programma chiamato
609C    LWPI        >83E0
60A0    MOV         ∂>702E, R11
60A4    B           *R11
60A6    AI          R1, >0008       incrementata R1 per continuare
                                    la ricerca del nome
60AA    JMP         >6074
60AC    LI          R0, >0F00
60B0    MOVB        RO, ∂>8322
60B4    LWPI        >83EO
60B8    B           ∂>6974
60BC    SZCB        ∂>76056, ∂>8349
60C2    LWPI        >7092
60C6    RTWP


65DA    LI          R1, >8008
65DE    AI          R1, >FFF8       R1-8
65E2    C           ∂>701E, R1
65E6    JEQ         >65EE
65E8    MOV         *R1, R0
65EA    JLT         >65F2
65EC    JMP         >65DE
65EE    INCT        R11
65F0    B           *R11
65F2    LI          R0, >0D00
65F6    B           *R11
6974    SOCB        ∂660D, ∂837C
697A    B           ∂>0070
```

Fig. 2 - La NAME LINK ROUTINE

MULTICOLOR

Chi ha letto gli articoli sull'assembler e L/M, saprà che il TI-99 ha quattro modi grafici, uno solo però è direttamente accessibile dal BASIC.
Il discorso cambia se al linguaggio a noi più familiare ne affianchiamo uno più familiare al nostro computer: l'assembler. Questo programma, scritto per la minimemory, è stato realizzato appunto, scrivendo dei sottoprogrammi in assembler che permettono di accedere facilmente alla grafica in multicolor.
Il programma assembler, di cui pubblichiamo il listato, è composto dai seguenti sottoprogrammi:
MULTIC, CLEAR, COLOR, LOAD
Le loro caratteristiche sono specificate nel riquadro a fianco.
Non ci dilunghiamo nello spiegare come il nostro TI-99 gestisce la grafica in multicolor, per chi volese saperne di più vi rimandiamo alla già citata rubrica sull'assembler, ci limitiamo solo a ricordarvi che in multicolor lo schermo viene diviso in 48-64 punti (o meglio quadretti) ognuno dei quali può assumere uno dei sedici colori disponibili.

IL PROGRAMMA

Il programma è stato registrato in due parti. La prima parte contiene il programma assembler e deve essere caricata selezionando il comando LOAD STORAGE FROM CSI dell'EASY BUG premendo il tasto «L». Effettuata questa operazione potete caricare il programma BASIC con il ben noto OLD CSI.
Caricato il programma potete finalmente dare il RUN.
Come prima cosa compare la tabella dei comandi disponibili. Questa tabella è richiamabile in qualunque momento premendo «CTRL 9» (HELP). Per comodità pubblichiamo nel riquadro a fianco la tabella dei comandi e **una mascherina da sostituire a quella originale del computer.** Sullo schermo in multicolor potete vedere uno sprite a forma di penna, questa vi dà due informazioni: la prima riguarda la posizione di scrittura, indicata dalla punta dello sprite, la seconda riguarda il colore di scrittura che è uguale a quello della cannuccia della penna. Potete spostare la penna sullo schermo usando sia la tastiera, con i soliti tasti «e, s, d, x» per le quattro direzioni e«w, r, c, z» per le diagonali, sia usando il Joystick, in entrambi i casi ricordatevi di sollevare il tasto ALPHA LOCK.
Durante gli spostamenti potete scegliere se scrivere o meno premendo il tasto shift insieme ai tasti o, se state usando il joystick, premendo il pulsante rosso. Il colore con cui state scrivendo può essere cambiato in qualunque momento premendo CTRL 7 seguito dal codice-colore. Il codice-colore è lo stesso della CALL COLOR,

Questa è la mascherina che dovete ritagliare, incollare su un cartoncino e sostituire a quella del computer. Vi consente di avere sempre sotto controllo il significato dei tasti funzione.

Colorare in rosso

Colorare in grigio

					ML
					CO
					multicolor

ABBONATEVI
A TI 99
NEWSOFT.
CON SOLE
L. 95.000
RICEVERETE
11 NUMERI
DELLA RIVISTA
E IN PIU'
AVRETE
DIRITTO AGLI
SCONTI SUL
SOFTWARE
E
HARDWARE
OFFERTO DA
TI 99 MARKET

Per abbonarsi
basta inviare
un assegno
non trasferibile
o un vaglia
postale a:

NEWSOFT SNC
Via S. Jacini, 4
20121 MILANO

FUNZIONI

Tasto	Funzione
CTRL 1	cancella il disegno in memoria
CTRL 6	cambia il colore dello schermo. Deve essere seguito dal codice colore.
CTRL 7	cambia il colore della scrittura. Deve essere seguito dal codice colore.
CTRL 9	Help
FCTN 2	fine del programma
FCTN 4	blocca il programma **DA NON USARE**
FCTN 6	carica un disegno in memoria
FCTN 8	salva il disegno in memoria

SOTTOPROGRAMMI ASSEMBLER

CALL LINK («MULTIC»)
Porta il computer in multicolor mode. Per tornare in graphic mode battere CALL LOAD (-31788,224) e ENTER, alla successiva pressione di un tasto si ritorna in graphic mode.
CALL LINK («CLEAR»)
Cancella lo schermo in multicolor
CALL LINK («COLOR», RIGA, COLONNA, CODICE-COLORE)
Colora il punto specificato dal numero di riga (0-47) per colonna (0-63), del colore specificato nel codice colore (1-16)
CALL LINK («SAVE»)
Salva sull'unità CSL il disegno presente in memoria
CALL LINK («LOAD»)
Carica un disegno in memoria da CSL
PRIMA DI UTILIZZARE QUESTI SOTTOPROGRAMMI IN UN PROGRAMMA BISOGNA ESEGUIRE UNA
CALL LOAD (-32658,14,0," ",-32732,13,248)
PER CREARE LO SPAZIO IN MEMORIA

COLORI

Codice	Colore
1	trasparente
2	nero
3	verde
4	verde chiaro
5	blu scuro
6	blu chiaro
7	rosso scuro
8	ciano
9	rosso
A	rosso chiaro
B	giallo scuro
C	giallo chiaro
D	verde scuro
E	nagenta
F	grigio
0	bianco

solo espresso in esadecimale; il bianco ha codice ∅. Per cambiare il colore dello sfondo bisogna premere CTRL 6 seguito dal codice-colore.
Le vostre opere in multicolor possono seguire due vie: la distruzione o il salvataggio su nastro.
Nel primo caso potete cancellare il disegno premendo CTRL 1. Se invece volete salvare il vostro disegno dovete premere FCTN 8, sul video compariranno le istruzioni per registrare su nastro il disegno. Se volete rivederli dovete premere FCTN 6.
Per terminare il programma non usate il tasto CLEAR ma FCTN 2 usando CLEAR infatti non si ripristina il graphic mode. Se per errore, durante l'esecuzione del programma, premete CLEAR, per riavere il disegno dovete seguire questa procedura:
battere CON e ENTER
date il comando di SAVE con FCTN 8
ora premete E ed ENTER e il vostro disegno è salvo.
Ci sembra che non ci sia altro da dire, vi auguriamo buon lavoro e inviateci i vostri disegni.

RICCARDO ROSSI

```
        B     *R11      TORNA AL BASIC


*       ***********************************
*       *     M U L T I C O L O R         *
*       *Programma per la gestione del     *
*       *MULTICOLOR MODE Version 1.1       *
*       *versione per  Ed.ASSEMBLER        *
*       *    Adapter : MONTINI EZIO        *
*       ***********************************

* NOMI DI ACCESSO AI SOTTOPROGRAMMI

        DEF SAVE,LOAD,MULTIC,CLEAR,COLOR

* NOMI DELLE SUPPORT UTILITY UTILIZZATE

        REF VMBW,VSBW,GPLLNK,VWTR,VSBR

*******************************************
* LE UTILITY SOTTO SPECIFICATE NON SONO   *
* PRESENTI NELL' ED.ASS., QUINDI PRIMA    *
* DI CARICARE IL CODICE OGGETTO DEL PRO-  *
* GRAMMA DA BASIC DOVREMO CARICARE IL     *
* PROGRAMMA  BSCSUP  CHE LE CONTIENE.     *
*******************************************

        REF NUMREF,XMLLNK

ST      BSS  32       BUFFER PREPARAZIONE SCHERMO

* COPIA DEL PAB DA COPIARE IN VDP

PA      DATA >0800,>0000,>0600,>6003

* NOME DELLA PERIFERICA SI I/O " CS1 "

T1      TEXT 'CS1'

        EVEN          PAREGGIA I BYTES

*  SOTTOPROGRAMMA "LOAD" PER CARICARE DISEGNO
* IN MULTICOLOR
*             CALL LINK("LOAD")

LOAD    LI    R1,>500   CODICE DSR LOAD
        JMP   DR

*  SOTTOPROGRAMMA "SAVE" PER SALVARE DISEGNI
*  IN MULTICOLOR
*             CALL LINK("SAVE")

SAVE    LI    R1,>600   CODICE DSR SAVE

DR      LI    R0,>07F0  INDIRIZZO PAB IN VDP
        BLWP  @VSBW
        INCT  R0
        LI    R1,PA
        LI    R2,11
        BLWP  @VMBW     SCRIVE LA PAB IN VDP
        LI    R0,>834A  PREPARATIVI PER ATTIVARE
*                       LA DSR ROUTINE DEL CS1
        LI    R1,T1
        MOV   *R1+,*R0+
        MOV   *R1,*R0

        LI    R1,3
        MOV   R1,@>8354
        LI    R1,>7FD
        MOV   R1,@>8356
        CLR   @>83D0
        MOVB  @>83D0,@>837C
        LI    R1,>0800
        MOVB  R1,@>836D
        LI    R0,>1E0
        BLWP  @VWTR     RIPRISTINA IL GRAPHIC MODE
        SWPB  R0
        MOVB  R0,@>83D4
        LI    R0,>400   RIPOSIZIONA LA TAVOLA DEI
        BLWP  @VWTR     PATTERNS

        BLWP  @GPLLNK   ATTIVA CS1
        DATA  >003D

*  SOTTOPROGRAMMO CHE PORTA LA GRAFICA IN
*  MOLTICOLOR MODE.
*             CALL LINK("MULTIC")

MULTIC LI     R0,>401
        BLWP  @VWTR     SPOSTA LA TAVOLA DEI
*                       PATTERNS
        LI    R0,>1EA
        BLWP  @VWTR     ATTIVA MULTICOLOR MODE
        SWPB  R0
        MOVB  R0,@>83D4 COPIA IL VALORE DEL REG.1
*                       IN >83D4
        CLR   R1        PREPARA LO SCHERMO
        CLR   R2
L1      MOVB  R1,@ST(R2) PREPARA IL BUFFER PER
*                        SCRITTURA IN VDP RAM
        AI    R1,>100
        INC   R2
        CI    R2,31
        JLE   L1
        CLR   R0
        LI    R1,ST
L2      LI    R3,4      SCRIVE NELLA MAPPA VIDEO
L3      BLWP  @VMBW
        A     R2,R0
        DEC   R3
        DATA  >0020
        JGT   L3
        CI    R0,>2E0
        JH    EXIT
        CLR   R3
L4      SWPB  R2
        AB    R2,@ST(R3)
        INC   R3
        SWPB  R2
        C     R3,R2
        JLT   L4
        JMP   L2
EXIT    B     *R11      TORNA AL BASIC

*  SOTTOPROGRAMMA "CLEAR" PER CANCELLARE VIDEO
*             CALL LINK("CLEAR")

CLEAR   LI    R0,>800   INDIRIZZO IN VDP RAM
        CLR   R1
LP      BLWP  @VSBW
        INC   R0
        CI    R0,>E00
        JLT   LP

        B     *R11      TORNA AL BASIC

*  SOTTOPROGRAMMA "COLOR" PER ACCEDERE AL VIDEO
*  IN MULTICOLOR MODE
*  CALL LINK("COLOR",RIGA,COLONNA,CODICE-COLORE)

COLOR   MOV   R11,R10   SALVA L' INDIRIZZO PER IL
*                       RITORNO AL BASIC IN R10
        CLR   R0
        LI    R1,1
        BL    @GN       LEGGE N. RIGA (0-47)
        MOV   @>834A,R8 R8=N. RIGA
        INC   R1
        BL    @GN       LEGGE N. COLONNA (0-63)
        MOV   @>834A,R9 R9=N. COLONNA
        INC   R1
        BL    @GN       LEGGE IL CODICE COLORE
        MOV   @>834A,R7 R7=CODICE COLORE
        DEC   R7        TOGLIE 1 PERCHE' CODICE
*                       COLORE 1 IN ASS.= 0
*                       CODICE 2 IN ASS.= 1 ECC.
        SWPB  R7
        LI    R6,>0F00
        SRA   R9,1      CONTROLLA SE IL NUMERO
        JOC   A         COLONNA E' PARI E SALTA
*                       SE NON LO E'
        SRC   R7,12
        SRC   R6,12
A       SLA   R9,3
        MOV   R8,R1
        LI    R2,8
        DIV   R2,R0     CALCOLA IL BYTE CHE DEVE
*                       COLORARE
        SLA   R0,8
        A     R1,R0
        A     R9,R0
        AI    R0,>800   R0=BYTE
        BLWP  @VSBR
        SZC   R6,R1
        SOC   R7,R1
        BLWP  @VSBW     COLORA IL PUNTO
        B     *R10      TORNA AL BASIC

GN      BLWP  @NUMREF   LEGGE IL VALORE PASSATO
        BLWP  @XMLLNK   DAL PROGRAMMA BASIC
        DATA  >1200     TRASFORMA IL VALORE IN
*                       NUMERO INTERO
        B     *R11      RITORNA AL SOTTOPROGRAMMA
*                       CHIAMANTE
        END
```

REALIZZIAMO UN ARCHIVIO SCACCHISTICO

Certamente una delle esigenze più sentite da parte di chi si occupa di scacchi ad un certo livello, è lo studio e l'aggiornamento continui. Perché non basta giocare ma bisogna anche saper trarre insegnamento dalle partite giocate da altri. Lo studio della teoria è fondamentale per un salto di qualità e costituisce l'elemento che distingue un dilettante che gioca per puro passatempo da chi nutre qualche aspirazione ed intende affrontare il gioco in modo meno improvvisato.
Sugli scacchi esistono molte pubblicazioni tra libri e riviste specializzate ed anche in alcuni quotidiani o settimanali di più ampia diffusione non manca la rubrica ad essi dedicata. Tuttavia non è improbabile che sia desiderio di molti gestire un archivio personale con una raccolta di partite catalogate secondo le aperture. L'impresa non è priva di difficoltà se non altro per l'eterogeneità delle fonti dalle quali si può attingere. E poi, la riproduzione di una partita richiede l'uso della scacchiera e il dover riordinare i pezzi per poter rivedere la stessa sequenza di mosse.
Ma è qui che gli scacchi possono parlare il computerese. Non ci riferiamo ai programmi per giocare a scacchi, che generalmente costituiscono più una curiosità che un effettivo banco di prova, ma ad un programma in grado di gestire in modo efficiente un archivio personalizzato creato dall'utente, capace di memorizzare un certo numero di partite e richiamarle in memoria per eseguirle sulla scacchiera che appare sullo schermo.
Il programma «Archivio scacchistico» è stato creato proprio per assolvere a questo scopo. Scritto in Extended BASIC è strutturato in modo professionale, tale da rendere più facile ed elastico il suo uso anche da parte di chi non ha molta familiarità con il computer. **La configurazione minima richiesta è data dalla console + video + registratore + modulo TI Extended BASIC** ma, per una maggiore efficienza, è raccomandabile un drive e, possibilmente, anche una stampante.
Il menu prevede sei opzioni: scrittura, modifica, memorizzazione, caricamento, esecuzione e stampa. Di queste le prime due meritano una particolare attenzione in quanto la buona riuscita del programma dipende in gran parte dall'esatta trascrizione delle mosse secondo regole ben definite. Infatti il programma ignora le regole del gioco ed esegue fedelmente tutte le mosse indicate anche se non sono ammesse, inoltre alcuni simboli sono diversi da quelli comunemente adottati ed in questo contesto assumono un significato particolare.
Veniamo subito alla notazione da usare. Ogni mossa completa comprende quella del bianco e quella del nero ed è contraddistinta da un numero progressivo. Per ogni mossa si utilizza una stringa di 12 caratteri, i primi sei per il bianco e gli ultimi sei per il nero. Considerando ognuna di queste due semistringhe di sei caratteri, il primo deve necessariamente contenere il simbolo del pezzo che muove, P=pedone, C=cavallo, T=*torre*, *A*=alfiere, D=dama, R=re. **Mentre nella notazione comunemente adottata, se il pezzo che muove è un pedone, non si indica, qui invece è obbligatorio**. Il secondo ed il terzo carattere sono le coordinate della casa di partenza mentre il quarto ed il quinto sono le coordinate della casa di arrivo. Le coordinate sono quelle standard, tuttavia esse vanno indicate esclusivamente con le lettere maiuscole. L'esatta individuazione delle coordinate e facilitata dalla presenza, anche in fase di scrittura, della scacchiera dove vengono eseguite le mosse in tempo reale. Si ribadisce che è necessario trascrivere sia la casa di partenza che quella d'arrivo contrariamente dall'uso più comune di indicare solo la casa d'arrivo. Nel caso dell'arrocco si useranno i simboli —0—0— per 0—0—0 per quello lungo. Il sesto carattere ha un significato dichiarativo ed è riservato per indicare particolari situazioni, come la presa di un pezzo, lo scacco al Re, la presa di un pedone «en-passant», la promozione, ecc. Per ognuna di esse si deve usare un simbolo convenzionale ben preciso. «:», i due punti indicano la presa di un pezzo; «+», il segno più significa lo scacco al Re; «*», l'asterisco indica la presa di un pezzo con contemporaneo scacco al Re; «%», il segno di percentuale si riferisce alla presa di un pedone en passant; «=», indica lo scacco matto. In assenza di una di queste situazioni particolari, il sesto carattere dovrà essere un trattino, «—» (segno meno). Nel caso della promozione di un pedone, il sesto carattere dovrà indicare il pezzo richiesto che, quasi sempre, è la dama. Per far ciò si useranno gli stessi simboli visti in precedenza, cioè «D» per la Dama, «T» per la Torre, «A» per l'Alfiere e «C» per il Cavallo. Resta da considerare come dire al computer che la partita è terminata. L'unico modo consiste nell'inserire, dopo l'ultima mossa, un messaggio con un punto come sesto carattere. In pratica si potrà scrivere: MATTO. , oppure: ABBAN. Anche altri messaggi funzioneranno altrettanto bene purché formati da cinque caratteri seguiti da un punto. Quando, infatti, il computer incontra un punto in sesta posizione, non interpreta la stringa come indicazione di una mossa ma come segno di fine partita.
Un esempio farà cadere le ultime perplessità. Riportiamo il famoso matto di Legal nelle due notazioni, quella standard e quella adottata nel programma.

1. e4, e5;
2. Cf3, Cc6;
3. Ac4, d6;
4. Cc3, Ag4;
5. C:e5, A:D;
6. A:f7+, Re7;
7. Cd5=matto.

1. PE2E4—PE7E5—
2. CG1F3—CB8C6—
3. AF1C4—PD7D6—
4. CB1C3—AC8G4—
5. CF3E5:AG4D1:
6. AC4F7*RE8E7—
7. CC3D5—MATTO.

Come si vede le due notazioni sono simili e quella utilizzata nel programma è contraddistinta da una maggiore rigidità. Vengono accettate solo lettere maiuscole ed ogni mossa consiste necessariamente di sei caratteri di cui il primo si riferisce al pezzo, quelli dal secondo al quinto costituiscono le coordinate ed il sesto è una particolare dichiarazione. Le mosse, scritte in questa forma particolare, vengono mantenute in memoria in una matrice unidimensionale,, sono interpretate dal computer durante l'esecuzione della partita sulla scacchiera presente sul video ed allo stesso modo vengono memorizzate su nastro o su disco. Tuttavia si può obiettare, non è questa la notazione alla quale è abituato il giocatore di scacchi. Ebbene, considerando questo il programma, prima di eseguire una partita, svolge una routine con la quale ricava una seconda sequenza di mosse traducendo quelle memorizzate nella notazione convenzionale. Il computer, pertanto, userà il primo come formato interno per memorizzare ed interpretare le mosse mentre usa il secondo solo per il display e la stampa, in modo da soddisfare più compiutamente le esigenze dell'utente.

COME UTILIZZARE IL PROGRAMMA

Supponiamo di voler trascrivere e memorizzare una partita. Selezionando il tasto 1 del menu, appare un'ulteriore richiesta: <N> per New e <A> per Append. L'opzione append si rivela utile quando siano già in memoria alcune mosse e si desideri completare la loro trascrizione senza dover iniziare tutto daccapo. Dovendo cominciare una nuova partita si premerà il tasto <N>.

I primi dati consistono nei nomi dei giocatori, che non possono superare i dieci caratteri ciscuno, luogo e data, per un massimo di venti caratteri, e tipo di apertura, per un massimo di venti caratteri, e tipo di apertura, per un massimo di sedici caratteri. Se le scritte dovessero superare le lunghezze consentite, l'utente dovrà effettuare delle abbreviazioni.

Dopo la fase preliminare si passa alla trascrizione delle mosse, facilitati dalla presenza della scacchiera sul video dove avviene la loro immediata esecuzione. I numeri progressivi vengono posti automaticamente e non resta che rispettare i simboli e la sintassi adottati dal programma scrivendo sempre ed esclusivamente con lettere maiuscole (abbassare il tasto ALPHA LOCK).
Il programma è in grado di rilevare errori fatali, come una stringa con meno di 12 caratteri, nel qual caso la mossa non viene accettata e deve essere riscritta. Quando si incappa in errori di altro tipo si possono scegliere due soluzioni.

1, proseguire riservandosi di correggere l'errore al termine, anche se cosi l'esecuzione sulla scacchiera risulterà compromessa.

2, uscire subito dal modo inserimento scrivendo un punto al termine della mossa corrente, come se fosse l'ultima. Questo secondo modo di procedere è da preferirsi. Tornati al manu principale, si passa alla modifica premendo il tasto 2. In questa sezione del programma non è più presente la scacchiera e le mosse compaiono a gruppi di venti. Nell'ultima riga appare un messaggio contenete i tasti di funzione abilitati: (BEGIN) = FCTN 5, (PROCEED) = FCTN 6, (AID) = FCTN 7, (REDO) = FCTN 8 = *(BACK)* = FCTN 9.

Per un uso facilitato dei tasti di funzione è opportuno ritagliare una striscia con le voci di riferimento ed inserirla nell'apposita guida sopra la tastiera del TI-99/4A. La funzione **BEGIN fa tornare al menu principale; PROCEED fa passare alla videata successiva contenente le venti mosse seguenti; AID fa apparire un segnale di prompt che andrà posto in corrispondenza dalla linea da correggere usando i tasti di controllo del cursore (FCTN E=alto, TCTN X =basso); REDO indica al computer che si vuole riscrivere la mossa; infine BACK fa tornare alla precedente videata.**

In pratica per corregere una mossa ci si porta sulla videata che la contiene con PROCEED e BACK, si preme AID per far apparire il prompt e lo si posiziona sulla riga da correggere, si preme REDO entrando in modo edit, si riscrive la mossa correttamente e si preme il tasto ENTER. Se non vi sono altre modifiche, si torna al menu con la funzione BEGIN e da qui si può proseguire la scrittura selezionando il tasto 1 e specificando il modo append che fa riprendere la trascrizione delle mosse dal punto in cui si era rimasti.

Terminata la scrittura e costatata la corretta esecuzione della partita sulla scacchiera, si inserirà un commento finale che, come è stato detto in precedenza, può essere MATTO o qualsiasi altra stringa purché formata da 5 lettere seguite da un punto. Il messaggio fa tornare al menu principale dal quale si può immediatamente passare alla sezione 5 per rivedere la partita.

Dopo una breve attesa viene riproposta la scacchiera; su di essa le mosse si succedono alla pressione di un tasto qualsiasi. La notazione scritta appare ora cambiata ed assume l'aspetto consueto che si ritrova nelle pubblicazioni dedicate agli scacchi. Durante questa fase sono abilitati due tasti di funzione: (BEGIN) = FCTN 5 per tornare al menu principale e (REDO) = FCTN 8 per ricominciare l'esecuzione della partita dall'inizio. La terza voce presente nel messaggio in fondo al video, cioè (KEY), indica che si deve premere un tasto per passare alla mossa successiva. A tal fine risulta più comodo il tasto SPACE.

Se si dispone di una stampante si può ottenere una copia su carta della partita (opzione 6 del menu), tuttavia, anche senza di essa, non viene sminuita l'efficacia del programma.

Per salvare la partita attualmente in memoria si deve premere il tasto 3 del menu e specificare poi il dispositivo a disposizione. CS1 è il registratore mentre DSK1 è il disk drive. La differenza sostanziale nell'uso dei due dispositivi sta nella gestione dell'indice delle partite e nelle modalità della ricerca. Con il registratore restano a carico dell'utente tutte le incombenze. Egli dovrà tener nota della posizione sul nastro di tutte le partite registrate, basandosi sui numeri riportati dal tape counter per poterle poi ritrovare con precisione. Nel caso si disponga del sistema a dischi tutto avviene automaticamente tranne la gestione dei «volumi». Ciò significa che resta a carico dell'utente solo la decisione sul dischetto da utilizzare ma una volta inserito egli disporrà di una gestione completa delle partite in esso contenute.

Per richiamare in memoria una partita bisogna specificare come in precedenza se si usa il registratore oppure il sistema a dischi. Con il primo si deve posizionare il nastro nel punto dove inizia la partita che interessa. Nel caso si usino i dischi appare invece l'indice delle partite con i nomi dei giocatori, luogo e data dove sono state giocate e tipo di apertura. Per caricare in memoria una partita è sufficiente digitare il numero corrispondente. In questa fase sono abilitati gli stessi tasti di funzione considerati in precedenza più la funzione ERASE = FTCN 3. Quest'ultima serve quando si voglia eliminare una partita precedentemente registrata. A tal fine si dovrà premere AID = FCTN 7, si sposterà il prompt sulla partita da eliminare e poi si premerà

GIOCHIAMO CON IL TI 99/4A

Un libro che non deve assolutamente mancare nella vostra libreria.

Potrete riceverlo a casa vostra a sole **L. 9500** + Spese di spedizione.

Per ogni richiesta di contrassegno, utilizzate il tagliando qui sotto spedendolo a:

NEWSOFT snc
via S. Jacini, 4
20121 MILANO

Vogliate spedirmi in contrassegno di L. 9500 + spese il libro: «GIOCHIAMO CON IL TI 99/4A»

Cognome .....

Nome .....

Indirizzo .....

.....

Cap. ..... Loc. .....

ERASE che significa cancella. Poiché la procedura è irreversibile il computer, prima di eseguire l'ordine, chiede una conferma. Con le altre funzioni è anche possibile correggere il contenuto dell'indice ma è bene sapere che in questo modo non si cambiano i dati generali che precedono ogni partita e la modifica rimane nell'ambito dell'indice.
Per non perdere la corrispondenza tra i dati dell'indice e quelli che precedono le partite è opportuno agire con cautela. Un modo di utilizzare queste funzioni di editing consiste nell'aggiungere a fianco del nome dell'apertura un particolare codice personalizzato per contraddistinguere ulteriormente le partite, codice che poi non comparirà sul video durante l'esecuzione.
Concludendo si può affermare che questo programma costituisce un valido strumento per chi voglia creare un efficace archivio scacchistico con la raccolta di intere partite o anche semplicemente di aperture. Rivedendo con facilità le varianti adottate dai campioni si approfondirà la conoscenza del gioco e si svilupperanno le proprie capacità di analisi.

AVVERTENZA PARTICOLARMENTE IMPORTANTE

Il programma occupa più di 14 Kbyte di memoria, cioè più di quanto sia consentito dal TI-99/4A in configurazione base con il modulo TI Extended BASIC inserito nella console. Il programma nella versione integrale non potrebbe venir registrato su nastro nemmeno con la Memory Expansion Unit. Infatti con l'espansione di memoria collegata i programmi Extended BASIC registrabili su nastro non possono superare i 12 Kbyte! Sembra un controsenso, ma espandendo il sistema i programmi Extended BASIC subiscono una contrazione; sono invece potenziati quelli memorizzati su disco che possono giungere oltre i 24K disponendo di altri 12K circa per le variabili per un totale di 36K.
Per non operare tagli drastici nel programma nato per sfruttare tutte le risorse del TI-99/4A ed allo stesso tempo soddisfare la maggior parte degli utenti che dispongono del solo registratore, si è adottata la soluzione di dividerlo in due parti.
La prima di queste consiste nel programma completo al quale sono state tolte solo le istruzioni riguardanti la gestione dei dischi e della stampante, la seconda, complementare alla prima, contiene solo le linee di programma che in precedenza erano state tolte.
Le linee che non compaiono nella prima parte sono: 1320-1390, 1500-2040 e 2570-2690. Inoltre sono state cambiate le linee 550-580, per eliminare la scelta n. 6 del menu per la stampa, e le 1310, 1530 e 1540 per sopprimere parti inutili o istruzioni GOSUB indirizzate alle linee mancanti.
Così chi ha solo il registratore utilizzerà unicamente la prima parte che è un programma completo ed autonomo, chi invece possiede anche il disk drive e l'espansione di memoria potrà utilizzare anche la seconda parte unendola alla prima con un'operazione di merging. A tal fine si riportano le istruzioni necessarie.

1, caricare dal nastro la prima parte e salvarla con il comando SAVE DSK1. PARTE1; 2, caricare dal nastro la seconda parte e salvarla con il comando SAVE DSK1. PARTE2, MERGE; 3, caricare nuovamente in memoria la prima parte con il comando OLD DSK1. PARTE1; 4, operare il merging con il comando immediato MERGE DSK1. PARTE2 (è necessario attendere un po' di tempo); 5, salvare il programma completo con SAVE DSK1. SCACCHI.

Chi usa il programma con i dischi, nella versione integrale necessita anche di un brevissimo programma per inizializzare il File Indice, programma che dovrà essere eseguito una sola volta prima di memorizzare la prima partita su un floppy disk. In seguito non dovrà più essere usato altrimenti si causerà l'azzeramento dell'indice delle partite!

SERGIO BORSANI

GESTIONE FILES SU CASSETTA

La scorsa volta abbiamo visto i modi di apertura del file, ora vedremo in che modo è possibile gestire (salvare e leggere) i dati, che abbiamo precedentemente scritto o che dobbiamo ancora scrivere.

Possiamo dividere le istruzioni dedicate a queste funzioni in due categorie:

a) scrittura e chiusura del file (PRINT, CLOSE)
b) lettura del file ..... (INPUT)

Categoria a

PRINT ÷numero del file che si vuole chiudere

Questa istruzione ha la funzione di scrivere i dati, numerici o alfanumerici, contenuti nelle variabili poste di dopo i due punti.

La lunghezza delle variabili non deve mai superare la lunghezza, FIXED, specificata in precedenza nella OPEN.

Pensiamo di avere un nastro, questo nastro è segnato con una riga ad ogni centimetro, noi dobbiamo scrivere dei dati su questo nastro, sappiamo che possiamo scrivere una lettera per casella oppure un numero in ognuna di esse.

Abbiamo un FIXED 64, corrispondente a 64 caselle di 1 cm. ciascuna.

I numeri occupano un solo quadratino, mentre le stringhe ne occupano tanti quanto sono le lettere o gli spazi vuoti che le compongono.

Deduzione, potremo scrivere 64 numeri oppure delle stringhe fino al raggiugimento del limite, o entrambe le cose.

Questo nastro prende il nome di record.

Ogni volta che eseguiamo una PRINT ÷n.: variabili, creiamo un record di lunghezza specificata, se le variabili non saranno sufficienti a riempirlo completamente, esso verrà completato automaticamente dal computer.

Scrivendo PRINT ÷1: A, B, C, B$, C$, sfrutteremo in maniera migliore il record, che creando tanti record quanti sono le variabili, PRINT ÷1: A ecc...

Ogni variabile rappresenta un campo all'interno del record.

Attenzione: un campo contenente una stringa non può essere letto da una variabile numerica e viceversa, le variabili devono essere sempre separate con una virgola.

CLOSE ÷numero del file che si vuole chiudere

CLOSE, come dice il mome ha la funzione di chiudere il colloquio con il dispositivo specificato nella OPEN.

Condizione necessaria per poter chiedere il file è che esso sia stato aperto in precedenza.

Per poter gestire nuovamente i dati appena scritti sul file, dovremo riaprire il file con una OPEN che abbia le stesse caratteristiche della precedente.

Categoria b

INPUT ÷numero file: lista delle variabili

INPUT, svolge una funzione inversa a quella svolta dall'istruzione PRINT.

Per poter leggere i dati dovranno essere verificate le seguenti condizioni:

— La OPEN deve avere le stesse caratteristiche di quella utilizzata per la scrittura dei dati (stesso FIXED, stesso tipo di scrittura dei dati, INTERNAL o DISPLAY ecc.).

— Il numero delle variabili, poste per leggere i dati, non deve mai differire dal numero dei campi del record che sono stati generati precedentemente.

— Come già detto in precedenza non è possibile leggere dati numerici con variabili alfanumeriche e viceversa.

— Le variabili devono essere separate da una virola.

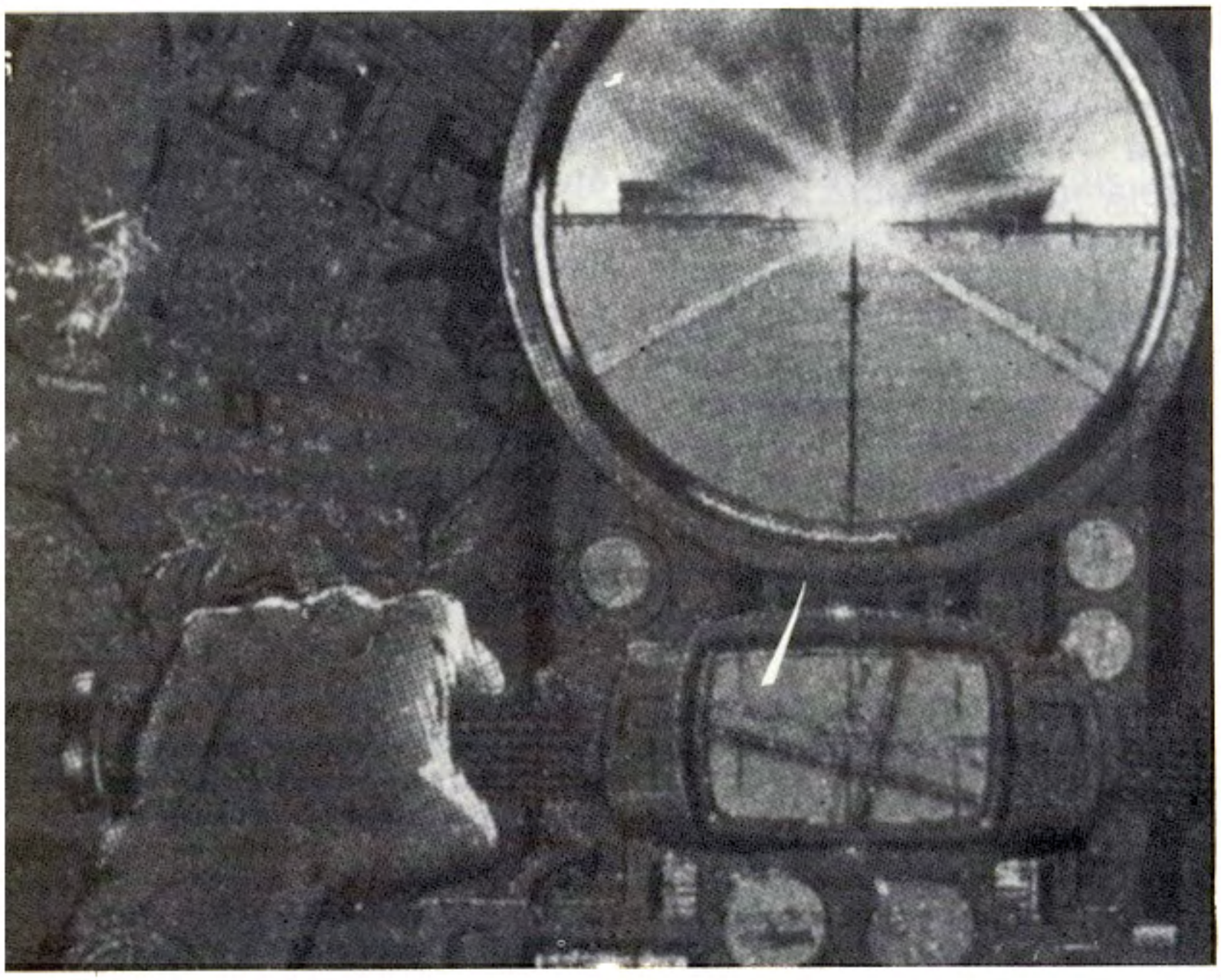
Ogni volta che viene eseguita una INPUT ÷n.: , il puntatore dei records viene incrementato, puntando al record successivo.

Questa annotazione perde di significato per il Files che usano come supporto magnetico il nastro delle cassette, perché il registratore avanza automaticamente ogni volta che il programma esegue una INPUT ÷n.: .

EZIO MONTINI

(2 - continua)

BATTAGLIA NAVALE

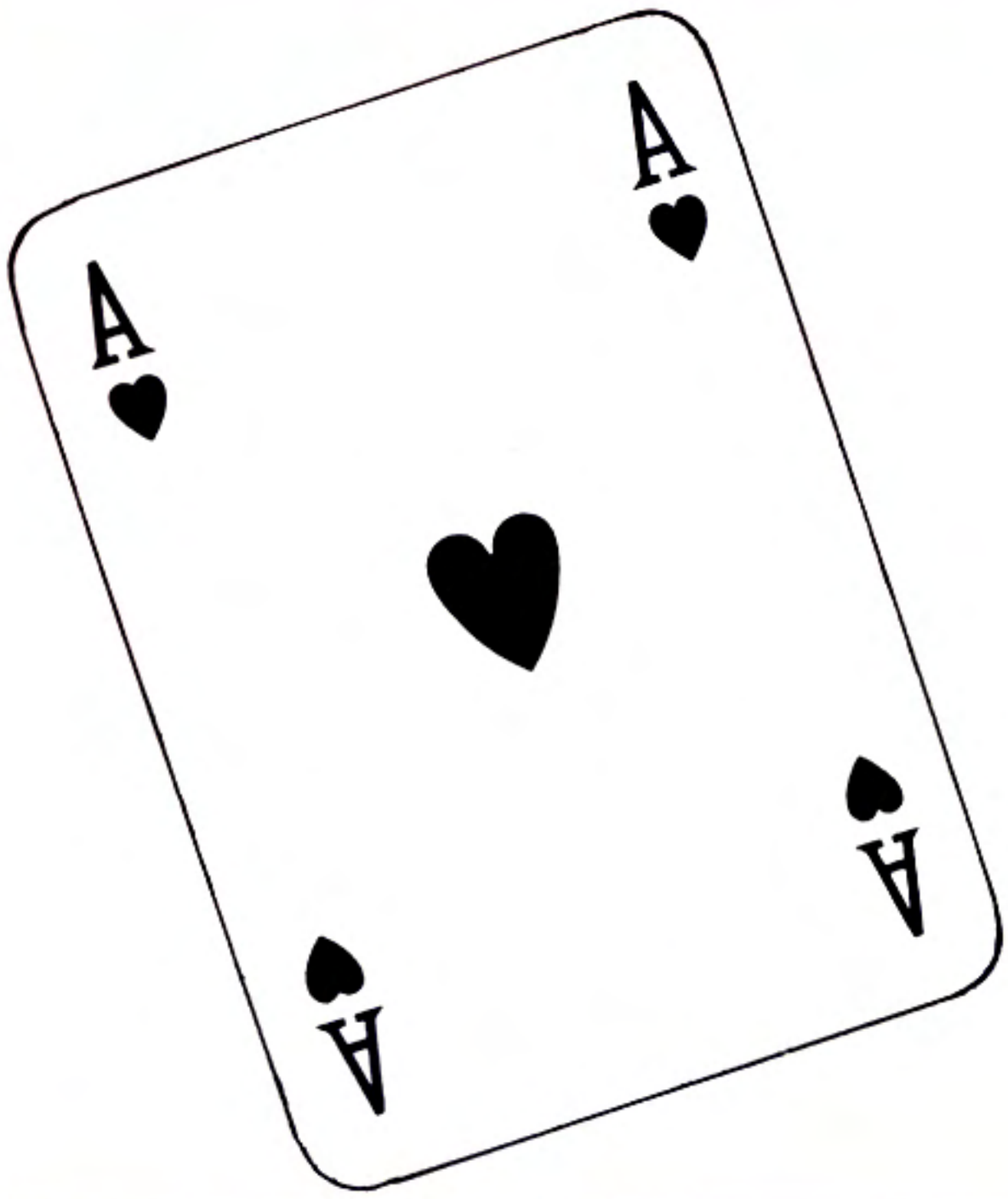


Con questo programma è possibile giocare a battaglia navale contro il computer, ad armi pari, scegliendo il numero e la lunghezza delle navi con le quali si vuole giocare.

NORME OPERATIVE

Il programma funziona in TI BASIC e richiede perciò la sola console.
Dopo aver lanciato il programma, alla richiesta "NAVE" rispondere digitando un numero compreso tra 2 e 10, ad indicare la lunghezza della prima nave, e premere ENTER.
Continuare poi con la seconda nave e le seguenti, tenendo presente che il massimo consentito è 30 come somma di lunghezze di nave e 10 come numero di navi.
Rispondere 0 quando si vuol terminare l'immissione prima della 10ª nave.
A questo punto sullo schermo comparirà la scritta ATTESA, e vi rimarrà sino a che non saranno state dislocate casualmente le navi scelte precedentemente in due matrici interne al programma.
Il computer chiederà poi chi deve tirare per primo, e dopo la risposta presenterà i due quadranti, in uno dei quali è raffigurato lo schieramento delle navi dello sfidante.
Il tiro del computer è automatico, mentre quello dello sfidante verrà effettuato premendo la barra spaziatrice dopo aver spostato opportunamente i 2 asterischi (cursori) con i tasti E (alto), X (basso), D (destra), S (sinistra), in modo da determinare le coordinate del tiro.
All'inizio della partita, alla destra del quadrante con le navi del computer, sono visualizzate tutte le navi in gioco; ad ogni affondamento da parte dello sfidante corrisponderà la scomparsa di una nave di pari lunghezza; in tal modo sarà possibile controllare la situazione delle navi rimaste.
Occorre tenere presente che due navi non possono toccarsi tra di loro, ma possono essere a contatto dei bordi del quadrante.

ALBERTO STRAFILE



POKER

Si tratta del classico, conosciutissimo gioco di carte, da giocare questa volta contro un abilissimo computer... senza paura tuttavia di perdere la camicia.
All'inizio il computer, dopo aver fatto la vostra conoscenza, distribuisce cinque carte scoperte a voi e cinque coperte a se stesso. Ricordate che come voi non potete leggere le carte del computer, così a lui non è permesso conoscere le vostre.
A questo punto il computer vi chiederà quante e quali carte volete cambiare.
Rispondete utilizzando soltanto i tasti da 1 a 5 (o lo 0 nel fortunato caso che siate serviti) e lui farà tutto il resto.
Ricordate che nel mazzo ci sono 28 carte (dall'otto all'asso); questo vi sarà utile nei progetti di scala.
Il computer scoprirà le proprie carte, come ogni buon giocatore, soltanto nel caso che lo abbiate battuto, ma vi assicuro che ciò accade abbastanza raramente!
Per la partita successiva premete un tasto qualsiasi e... buona fortuna!

LEVIO PEROTTI

ANALISI DEL LISTATO

1 - 11	Titolo, definizione caratteri
12 - 13	Assegnazione colori
14 - 16	Assegnazioni variabili
17 - 20	Azzeramenti - Scelta delle carte
21 - 24	Distribuzione delle carte
25 - 70	Routine con cui il computer decide quali e quante carte cambiare
71 - 105	Routine cambio carte del giocatore
106 - 110	Scopre le carte del computer in caso di vincita
111 - 113	Call Key per la partita successiva

Sottoprogrammi

114 - 118	Subroutine per il cambio delle carte
119 - 122	Estrazione e controllo delle carte
123 - 130	Disegna la carta bianca
131 - 132	Disegna la carta coperta
133 - 182	Assegna il seme, il colore e disegna la carta relativa
183 - 187	Legge il punteggio in mano
188 - 193	Riconosce il poker e il colore
194 - 203	Riconosce il tris
204 - 213	Riconosce la coppia
214 - 220	Riconosce la scala mancata
221 - 228	Riconosce il valore del poker
229 - 238	Riconosce il full
239 - 278	Riconosce la doppia coppia
279 - 326	Riconosce il valore del full, tris, d. coppia, coppia
327 - 330	Riconosce la scala reale
331 - 343	Riconosce la scala
344	Scrive i punteggi

UN ARCHIVIO SU MISURA
DATA BASE TI 99 4/A

Il programma scritto in Extended Basic, risulta estremamente flessibile ed adattabile a qualsiasi tipo di piccolo archivio o alla memorizzazione di testi.
L'unica vera limitazione, per chi non possiede l'espansione di 32 K, è la limitata memoria del computer, e la lentezza dei trasferimenti dati con il registratore (superabile dai possessori del sistema a dischi).
Il programma lascia liberi circa 4800 Bytes per i file, sufficienti per un centinaio di piccole schede.
Qualora ne occorressero di meno, si può modificare in riga 20... l'istruzione
DIM I$ (100, 7).
— lo stesso dicasi per il numero di campi per ogni scheda, dimensionato nella stessa riga a 7
— diminuendo opportunamente la DIM... si recupera memoria, come pure — se non se ne prevede l'utilizzo — si può eliminare la routine di ordinamento alfabetico, o di indice, che d'altra parte risultano già interdette memorizzando un testo.

L'opzione «ELIMINARE» può essere eliminata sfruttando opportunamente l'opzione «MODIFICARE».
Penso però che non convenga mutilare il programma, poichè viste le applicazioni prevalentemente di tipo casalingo e familiare verso cui per forza di cose si rivolge l'utente del 99, la diposnibilità di memoria è sufficiente per la maggior parte di esse.

N.B. L'effetto monitor, qualora non gradito, è eliminabile cancellando parte della riga 20. Attenzione a non cancellare DIM...
Un cenno particolare merita la routine di ricerca, sfruttata anche per la modifica.
La ricerca vera e propria viene effettuata in riga 670 sfruttando la funzione POS (I$(M, N), Q$, 1).
Tale funzione restituisce il valore 0 se non si è trovata Q$ in I$ (M, N), altrimenti restituisce la posizione del primo carattere di Q$ in I$ (M, N), con tale funzione si può ricercare una parola o una lettera in qualunque posizione si trovi.
Qualora si volesse ricercare una parola che stia solo all'inizio di un campo, sostituire in riga 670 la istruzione IF POS (I$ (M, N), Q$, 1) = 0 THEN 780 con IF SEG$ (I$ (M, N), 1, LEN (Q$) <> Q $ THEN 780.

DESCRIZIONE DEL PROGRAMMA

Qui di seguito troverete la spiegazione dei vari punti che compongono il menù del programma.

1 RICHIAMARE
2 CREARE O AGGIUNGERE
3 ORDINARE ALFABETICAMENTE
4 INDICE
5 VISUALIZZARE
6 RICERCARE
7 MODIFICARE
8 ELIMINARE
9 REGISTRARE
0 FINE

1. Richiamare

Qualora vi sia già un archivio in memoria, ciò viene evidenziato, viene presentato il titolo e si ritorna al quadro principale. In caso contrario, una volta letto il primo record, viene presentato sullo schermo il titolo dell'archivio ed altre informazioni relative ad esso; segue la richiesta se procedere o no alla lettura. Ciò per evitare errori di lettura di cassette, avendone varie con vari tipi di archivio, e data la lentezza della procedura di lettura dal registratore.
In fase di lettura viene evidenziato sullo schermo il numero del record che si sta leggendo, ed il numero totale dei record costituenti l'archivio.

2. Creare un archivio o aggiungere record

INPUT «TITOLO»
INPUT «TESTO o SCHEDARIO»

↓

INPUT «Numero di campi per ogni Record» (max 7, per contenere intero record in una videata, teoricamente è possibile aumentare il n. dei campi)

↓

INPUT «VUOI CREARE UNA MASCHERA?» (possibilità di correggere la maschera prima di farla accettare, controllo automatico della lunghezza della maschera con indicazione di quanti caratteri sono eventualmente eccedenti)

↓

INPUT RECORD con presentazione della eventuale maschera, (controllo automatico della lunghezza del record ed eventuale indicazione del numero di caratteri eccedenti) - (eventuali correzioni possono essere fatte successivamente con l'opzione «MODIFICARE»).
Per terminare introduzione battere: «END».

N.B. Qualora vi sia già un archivio in memoria, perché richiamato o creato, tale opzione dà luogo all'aggiunta di nuovi record all'archivio in memoria.

3. Ordinare alfabeticamente

INPUT: «Secondo quale campo vuoi ordinare?»
L'ordinamento alfabetico può essere effettuato in qualsiasi momento.

N.B. Tale opzione risulta interdetta per i file: «TESTO».

BREVE ANALISI LISTATO

20	Effetto monitor, dimensionamento matrice per memorizzare record
30- 60	Stampa quadro principale
70- 80	Scelta e invio alla routine prescelta
90- 220	Routine RICHIAMARE
230- 420	Routine CREARE
430- 580	Routine VISUALIZZARE
590	Inizio routine MODIFICARE fino a 860
600- 860	Routine RICERCARE
870- 980	Routine ELIMINARE
990	Visualizzazione messaggio di archivio vuoto e inizio routine REGISTRARE fino a 1130
1140-1200	Routine FINE
1210-1230	Routine CALL KEY
1240-1360	Routine ORDINARE ALFABETICAMENTE
1370	Controllo se archivio è un testo per interdire INDICE e ORDINE ALF.
1370-1430	Routine INDICE

PRINCIPALI VARIABILI USATE

J	= Numero di record
X	= Numero di campi
P	= INT (21/X) variabile usata per inquadramento automatico del record sullo schermo
NOM $	= Titolo archivio
MAS $ (X)	= Maschera
I $ (J, X)	= Matrice contenente i campi

VARABILI DI STATO

C = 1	Se si è richiamato o creato un archivio
S = 1	Se si è creato un archivio
T = 1	Se l'archivio creato è un testo
MOD = 1	Se si sta usando l'opzione «MODIFICA»
MO = 1	Se si è modificato un record
E = 1	Se si è eliminato un record
R = 1	Se si è registrato l'archivio in memoria
MA.S = 1	Se si usa una maschera
TR = 1	Se si ricerca si è trovato un record richiesto.

4. Indice record

INPUT «Secondo quale campo vuoi l'indice?»
È possibile avere in sequenza di 10 in 10 l'indice dei record secondo qualsiasi campo.

N.B. Tale opzione è interdetta per i file: «TESTO».

5. Visualizzare

Inquadramento automatico sullo schermo, dei record richiesti, secondo il numero di campi.

2. Opzioni:

1 - Vedere tutto l'archivio
vengono presentati in sequenza tutti i record, con possibilità, in ogni momento, di tornare al quadro principale
2 - Vedere un determinato record
specificandone il numero

6. Ricercare

È sufficiente introdurre — a richiesta del computer — una lettera, una parola o un intero campo, del record da ricercare.
Verranno presentati in sequenza tutte le schede contenenti, in qualsiasi posizione, la stringa digitata. Naturalmente più è lunga la stringa ricercata, più la ricerca sarà precisa e mirata.
— esaurita la presentazione dei record trovati, si ritorna al quadro principale
— l'eventuale ricerca infruttuosa viene evidenziata
— vi è la possibilità, dopo ogni scheda presentata, di tornare al quadro principale.

7. Modificare

Conoscendo il numero della scheda da modificare si accede direttamente al record richiesto, altrimenti si può procedere come per la ricerca, digitando una stringa contenuta nel record da modificare.
Viene presentato il record campo per campo con la richiesta «vuoi modificare? (S/N)».

8. Eliminare

Viene richiesto il numero del record da eliminare. Viene presentata la scheda corrispondente e viene chiesta conferma se procedere o no alla eliminazione.
A risposta affermativa la scheda viene cancellata e l'archivio viene automaticamente ricompattato.

9. Registrare

Se non c'è nessun archivio in memoria, ciò viene evidenziato e si ritorna al quadro principale. Viene trasferito l'archivio su cassetta, durante l'operazione lo schermo riporterà il titolo dell'archivio, il numero totale dei record ed il numero di record che via via viene registrato.

0. Fine

Trattasi di una fine logica del lavoro infatti qualora si siano effettuate operazioni di
— CREARE O AGGIUNGERE
— MODIFICARE
— ELIMINARE
il computer avverte circa l'operazione effettuata e chiede se si vuol registrare il nuovo archivio.
A risposta affermativa si attiva la rutine di registrazione, altrimenti termina il programma.

MARIO AGAMENNONE

ABBONATEVI
A TI 99
NEWSOFT.
CON SOLE
L. 95.000
RICEVERETE
11 NUMERI
DELLA RIVISTA

E IN PIU'
AVRETE
DIRITTO AGLI
SCONTI SUL
SOFTWARE
E
HARDWARE
OFFERTO DA

TI 99 MARKET

Per abbonarsi
basta inviare
un assegno
non trasferibile
o un vaglia
postale a:

NEWSOFT SNC
Via S. Jacini, 4
20121 MILANO

PROVA DI MEMORIA

Un giochino semplice ma psicologicamente valido per mettere alla prova la capacità di memoria a corto termine dei lettori. Il computer propone per un breve istante un numero da 0 a 99. Quando il numero scompare, occorre ripetere lo stesso numero battendolo sulla tastiera e premere enter. Dopodichè il computer mostra una sequenza di due numeri da ricordare. I due numeri vanno ripetuti sulla tastiera nello stesso ordine di apparizione. Se non si commettono errori, i numeri mostrati passano a 3 e così via di seguito in progressione. Presto si scopriranno paurosi... vuoti di memoria, e si commetterà un errore. Poco male: fino a due errori sono ammessi. Ad ogni errore però il computer prosegue mostrandovi una sequenza con un numero minore di elementi, per darvi la possibilità di recuperare. Al terzo errore, il gioco finisce e viene mostrata la vostra migliore prestazione. Sembra facile, ma non lo è. Un livello superiore a 6 è difficilmente raggiungibile. Un pizzico di fortuna è anche necessario. Basta ottenere una sequenza lunga ma facile da ricordare per realizzare un punteggio imbattibile. Il programma può anche esser utilizzato per esercitare e migliorare la capacità mnemonica, o per sperimentare metodi di memorizzazione. Pico della Mirandola avrebbe senz'altro realizzato punteggi elevatissimi. Per fortuna, non aveva a disposizione un computer.

GEY

DESCRIZIONE DEL PROGRAMMA

Linee	Descrizione della funzione
90- 240	Inizializzazione e creazione schermo
260- 415	Proposta della sequenza di numeri
420- 442	Accettazione risposte
443- 470	Elaborazione delle risposte
1000-1020	Ripetizione della prova di memoria
9000-9010	Aggiornamento risultati

THE CAVE

Questo programma è stato scritto in TI BASIC. Il programma gira però anche in EXTENDED BASIC, ma è consigliabile utilizzarlo solo da TI-BASIC, perché in EXTENDED è troppo difficoltoso giocare, ed è molto arduo riuscire a superare il primo schema.

Ogni schermo è caratterizzato da numerosi piani, collegati tra loro da altrettante scalette. Il compito del nostro eroe è quello di raccogliere i picconi sparsi qua e là sui vari piani. Purtroppo, per lui, la cosa non sarà tanto semplice, in quanto numerose anfore semoventi faranno di tutto per schiacciarlo. Inoltre, dovrà sempre fare attenzione a non sfracellarsi da un piano all'altro.

Quando il nostro eroe sarà riuscito a impossessarsi di tutti i picconi, dovrà portarsi in basso a destra dello schermo a perforare i due muri che si trovano. Così facendo, si passerà allo schermo successivo di difficoltà crescente e via di seguito. Un'ultima raccomandazione: avete solo due omini, quindi servitevene con prudenza e parsimonia e BUON DIVERTIMENTO A TUTTI.

GIAMPAOLO BOTTINI
DANIELE CATALFAMO

ANALISI DEL LISTATO DI THE CAVE

100- 290	definisce caratteri e colori
300- 750	dopo la schermata di presentazione inizializza le variabili
760- 1130	stampa lo scenario del gioco
1140- 1210	data per stampare lo scenario del gioco
1220- 1290	inizializza altre variabili
1300- 1590	loop principale
1300- 1350	controlla quali tasti sono stati premuti
1360- 1450	muove l'omino in verticale
1460- 1590	muove l'omino in orizzontale
1600- 1720	fa saltare in alto l'omino
1730- 2010	se è stato premuto il tasto S o D dopo essere saltato in alto, l'omino si sposta rispettivamente a destra e a sinistra
2020-	se la caduta si prolunga troppo, l'omino cade e perde una vita
2030- 2050	a seconda della posizione dell'omino fa muovere i nemici più vicini
2060- 2510	routine di movimento nemici
2520- 2670	routine per l'abbattimento dei muri
2680- 2730	routine per la presa di piccone
2740- 2870	routine che viene richiamata quando l'omino perde una vita
2880- 3040	routine di fine quadro
3050- 3190	routine di stampa punteggio e fine gioco
3200- 3450	routine di presentazione e istruzioni
3460- 3550	data per i caratteri grandi

IL COEFFICIENTE DI CORRELAZIONE R

Sovente in satistica, ma direi anche nella vita quotidiana, è necessario conoscere il grado di mutua dipendenza tra due serie di dati, come dire, la correlazione tra causa ed effetto.
Per la conoscenza di questo fenomeno ricorriamo spesso al calcolo del coefficiente di correlazione R, detto anche di Pearson, per mezzo del quale è possibile sapere se esiste una relazione lineare tra le serie di dati in esame.
Il valore numerico di R sarà uguale a 0 nel caso in cui è da escludere la possibilità di correlare due fenomeni, sarà uguale a 1 nel caso in cui esista un relazione diretta mentre, di converso, diverrà uguale a -1 nel caso la relazione sia inversa (all'aumentare di un fenomeno corrisponda la diminuzione di un altro).
Con l'aiuto del programma allegato è possibile verificare la mutua dipendenza di variabili, posto che esse siano quantizzabili numericamente, e prendere pertanto decisioni logiche con l'aiuto del calcolatore.
A qualcuno dei più attenti sorgerà un dubbio:
Bene, e se questi valori ci daranno un valore di R tale per cui troveremo una correlazione anche nel caso in cui i valori siano del tutto dovuti al caso?
Risposta: non ci stancheremo di ripetere che la statistica non è una scienza fine a sè stessa ma, anzi, essa è destinata a guidarci, razionalmente, nel prendere decisoni, per quanto possibile, immuni dalla casualità.

APOLOGO

Facciamo un esempio:
Un giorno, per avventura, traversando la strada, per due volte (fatalità), ci imbattiamo camminando, su dei residui di quel nobile animale amico dell'uomo e poi, spinti da istinto profetico — dicono che porti bene — comperiamo due biglietti della lotteria parrocchiale e poi, è il colmo, vinciamo.
Incassato la vincita, e riparate le scarpe, occorre assumere un atteggiamento razionale e attento all'evolversi degli sviluppi futuri.
Che si fà?

1ª opzione.
Corriamo a casa, accendiamo il TI 99, furiosamente ci mettiamo a cercare un programma di statistica che, ci sembra, stia sotto il tavolino, in cantina, per reggere un piede malfermo. Facciamo girare il programma e, dopo qualche impropero, stante la difficoltà nell'immettere i dati, scopriamo un'ottima correlazione tra il fatto di avere pestato i nobili resti dell'amico dell'uomo e quello di avere vinto la lotteria.
Usciamo e ...
Narrano alcuni prevosti di campagna di aver notato, verso la mezzanotte, strani tipi, dallo sguardo un pò strano, che dopo aver bussato in canonica, ed essere entrati con un sorriso un pò ambiguo e con scarpe dal profumo indecoroso, hanno fatto incetta di biglietti della lotteria (il 1° premio era costituito da un puffo alto un metro e mezzo).

2ª opzione.
Premete ENTER e calmatevi; cercate in casa il programma di NewSoft. (Si questo della correlazione, il programma di statistica che sta in biblioteca).
Accendete una sigaretta, se fumate, se non è meglio, e sorseggiate un bicchiere di cognac dopo averlo ben scaldato tra le mani (a proposito avete rinnovato l'abbonamento? Siamo sotto le "feste" fate presto, sconti per gli affezionati lettori e per le comitive).
Bon, fate girare il programma. Esiste correlazione tra i due eventi? Si! Bene, ... ma l'evento è del tutto causuale ... male.
Morale. **
Non fa nulla, così van le cose del mondo; invece di spende-

1985

Sarà un anno di grosse e importanti novità per chi è rimasto fedele al buon TI-99.

«Novantanoviani» continuate a seguire con «TI-99 NEWSOFT» tutti gli sviluppi di questa «NEVERENDING STORY» che si chiama TI-99/4A.

DA TUTTA LA REDAZIONE BUON ANNO E... BUON LAVORO CON IL VOSTRO TI-99

re soldi in lotterie, concorsi, giochi delle tre tavolette, estrazioni a sorte e riffe varie, la statistica ci insegna che conviene non fidarsi; risparmiamo e spendiamo i soldi per un viaggio alle Seyschelles.

**Nota filosofica

Possiamo definire la statistica come quella scienza che attraverso equazioni matematiche permette di risparmiare le scarpe dai cattivi odori procurando, nel contempo, una acuta osservazione delle cause che li hanno generati.

LO — SI — TOH
(pensatore orientale)

Dove eravamo rimasti? A si. Al fatto che il nostro programma permette di separare la casualità dai fatti oggettivi.

Ricordate nel numero precedente il test T di Student e quelle frasi? : «La probabilità che tale evento sia avvenuto per caso è ...» anche nel test di correlazione viene eseguito un test di probabilità.

I messaggi assegnano le probabilità nel modo seguente:

0% - 1%	Esiste una forte evidenza che esista una relazione diretta tra X e Y
1% - 5%	Esiste una certa evidenza per supportare l'esistenza di una relazione tra X e Y
5% - 95%	Non esiste alcuna evidenza
95% - 99%	Esiste una certa evidenza per una relazione inversa tra X e Y
99% - 100%	Esiste una forte evidenza per una relazione inversa tra X e Y

Provate a far girare il programma applicandolo alla vita quotidiana (per i lettori più giovani: non usatelo per esaminare i vostri voti scolastici nel caso non abbiate studiato a sufficienza; non serve), prenderete confidenza con uno degli argomenti più importanti della statistica utilizzando il vostro computer come una macchina decisionale.

Ed è anche per questo che è stato creato il TI 99/4A.

Note al programma:

È permesso, per facilitare le risposte, rispondere alle domande usando sia i caratteri maiuscoli che quelli minuscoli.

Introducete prima il numero totale delle coppie di dati-esso dovrà essere compreso tra 2 e 100.

Immettete poi le coppie di dati X,Y separandoli con una virgola, premete ENTER e così via.

Nel caso siano stati commessi degli errori, è data la possibilità, nel corso del programma, di cambiare i dati, aggiungerne di nuovi o semplicemente listare i dati per il controllo.

Importante, infine, la conversione del coefficiente di correlazione R alla t di Student per mezzo della quale calcoliamo la probabilità che i dati siano casuali.

La formula usata è:

$$t = \frac{Rx\ \sqrt{\text{Numero dei gradi libertà}}}{\sqrt{(1 - R \times R)}}$$

dove il numero dei gradi di libertà è dato dal numero di coppie di dati meno 2.

PAOLO CIVARDI

TROVARE L'INTROVABILE

Sono un lettore della rivista TI99NEW SOFT, e avendo visto il vostro annuncio per l'ESPANSIONE DA 32K, ho pensato di scrivervi per chiedere delle informazioni a me molto utili essendo un neofita nel campo dei computers.
Io sono possessore di un TI 99/4A, e avrei il desiderio di aumentarne la capacità di memoria ram, aggiungendo quindi l'ESPANSIONE DA 32K di vostra produzione, ma nel vostro annuncio c'è scritto che per accedere a questa memoria esterna ci vuole l'EXTENDED BASIC.
Io non essendo in possesso dell'EXTENDED BASIC, volevo chiedere se c'è qualche altra soluzione, o se c'è una ditta o un negozio il quale venda questo introvabile EXTENDED BASIC.
Grazie.

CORRA' VASCO
Schio

Spett.le Newsoft
desidero ringraziarvi per l'aiuto fondamentale dato agli sfortunati possessori del nostro TI 99.
La rivista è giunta in un momento direi quasi di sconforto e che ha contribuito non poco a risollevare il morale di noi tutti.
Ottimi i programmi, ottima la spiegazione buono il prezzo.
Però avrei alcune proposte da fare. Ho visto che mettete in vendita programmi che però richiedono anche accessori che non sono in possesso di tutti e, per di più, neanche in commercio (tranne l'espansione da 32 K da voi messa in vendita).
Questo succede in Italia, mentre so che negli USA c'è tanto di quel Software e Hardware per il TI 99 da far impallidire anche i Commodoriani. E qui la mia proposta: perché la "NEWSOFT" non si fa promotrice per portare a conoscenza degli interessati dove trovare e acquistare materiale per il TI 99 (io ad es. non riesco a trovare la RS 232)?
E inoltre: perché non fare gli acquisti tramite Voi cercando così ognivolta di spuntare sulla quantità il prezzo migliore?
Sicuro di un vostro interessamento.

VECCIA MAURIZIO
Castel d'Azzano (VR)

Finalmente qualcosa anche per noi del TI/99 grazie e complimenti.
Vista l'attuale difficoltà a reperire materiale per il TI/99 (Extended Basic - Mini Memory ecc.) data dal listino della Texas dal mercato degli Home, vi chiedo se potreste indicarci su uno dei prossimi numeri, indirizzi di negozianti o magazzini a lui rivolgersi per l'acquisto di materiale per poter ampliare il nostro TI/99.
Fiducioso in un vostro interessamento cordialmente vi saluto.

GALIMBERTI ALBERTO
Pavia

Rispondendo agli accorati appelli di richiesta di periferiche ed espansioni di molti dei nostri lettori, siamo lieti di presentare un listino completo di hardware e software per il Texas TI 99. I più attenti avranno comunque già notato che questa felice innovazione è presente dal numero di Dicembre.

MS

MINI MEMORY E CS2

Posseggo un TI 99/4A ed il modulo SSS "MINI - MEMORY".
Desidererei avere direttamente o tramite la vostra rivista informazioni circa l'uso di questo strumento e circa le procedure ed i comandi da fornire per entrare ed utilizzare gli elementi da esso messi a disposizione (assembler, memorizzazione, ecc.).
Nonostante i molti tentativi da solo non ci sono riuscito.
Vorrei inoltre sapere come fare per collegare un secondo lettore/registratore di cassette.
Vi ringrazio fin da ora per la vostra cortese risposta.

Riguardo alla prima domanda possiamo solo rispondere: abbia fiducia e continui a seguire la nostra rivista, un passo alla volta affronteremo tutti i problemi. Per collegare

una seconda unità al computer è necessario avere il doppio cavo di interfacciamento della Texas che però, badi bene, è abilitato alla sola registrazione e non lettura dei dati.

MS

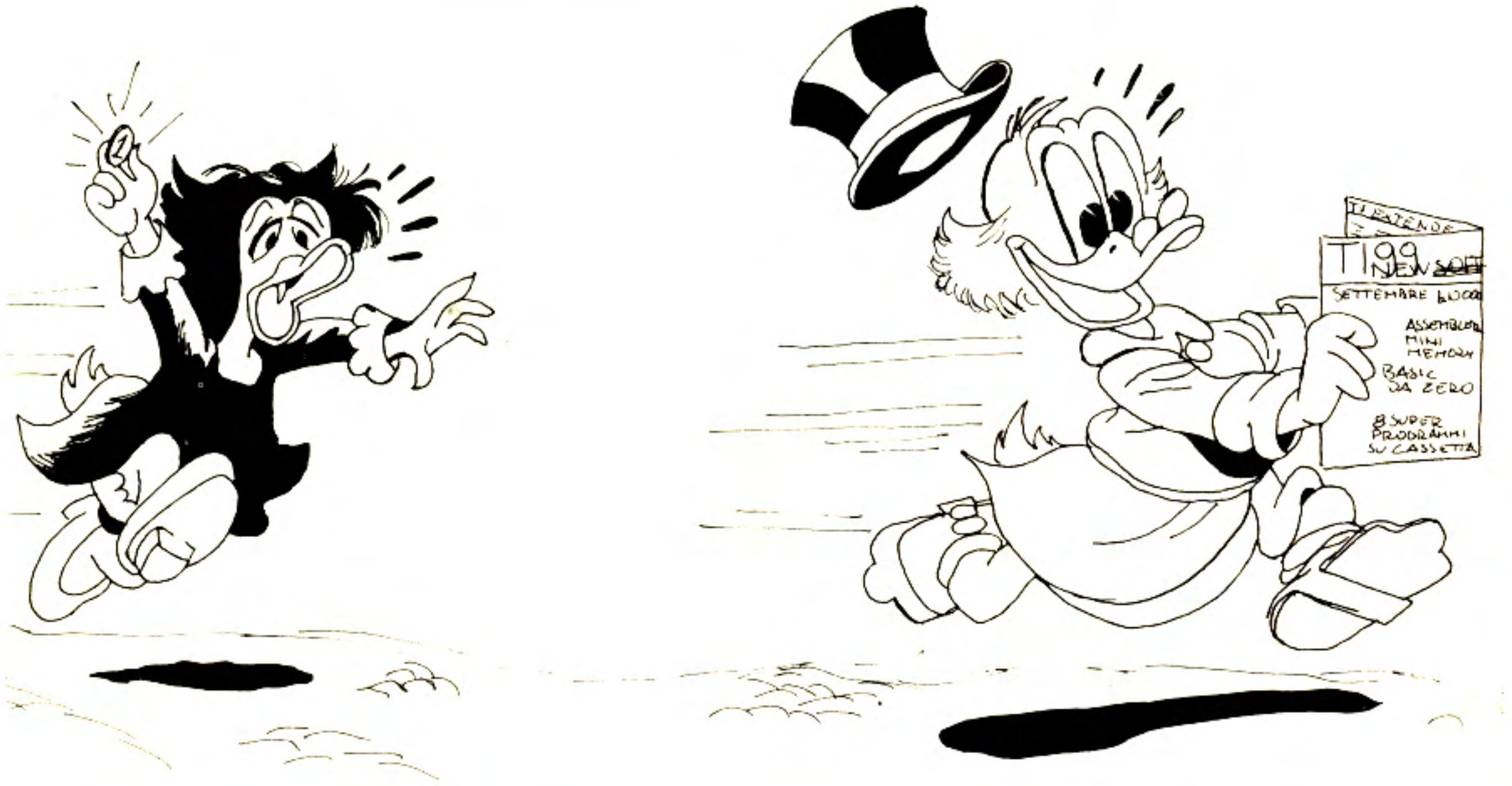
IL REGISTRATORE FA I CAPRICCI

Sono in possesso di un registratore Sanyo che possiede la presa per microfono, cuffia e start. Collegate le tre spine del computer, il registratore non entra in funzione su comando da tastiera, ma sono costretto a staccare la spina jack dello start ed avviare manualmente.
Come posso risolvere il problema? E' necessario intervenire sul registratore oppure non mi resta che acquistare il registratore originale Texas? Gradirei una risposta. Grazie.

ZAGHI NAZZARENO
Milano

Il problema che sembra così drasticamente concludersi con l'acquisto di un altro registratore è invece immediatamente risolvibile con l'introduzione di un adattatore alla presa dello start. E' uno spinotto grigio e nero che veniva dato in dotazione all'acquisto del cavo completo e che invertiva la polarità degli impulsi. Pensiamo che si possa ancora trovare con una certa facilità in circolazione.

MS



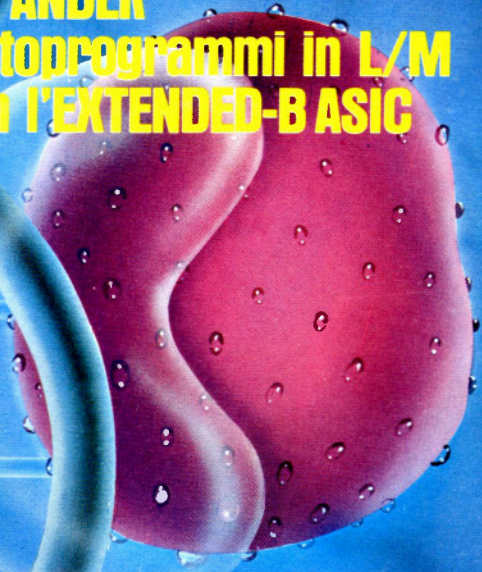
Questo mese premiamo due vignette.
La prima la trovate nella cassetta ed è di ANNA BIGATI di Genova.
L'altra è di FABRIZIO DE LEO di MARINO (Roma).
A tutti e due un abbonamento omaggio per 11 numeri a TI-99 NEWSOFT.

FEBBRAIO 1985 L. 10.000

TI 99 NEWSOFT

EXTENDED + 32 K
Programmiamo in Linguaggio Macchina

EXPANDER
Sottoprogrammi in L/M con l'EXTENDED-BASIC



MERGE per Mini Memory

IN SOFTWARE & SOFTWARE:
EXPANDER - Merge - Rockfeller
Lunar lander - Calorie
Musical Aider...

SOMMARIO

□ **TI BASIC**
Gestione files su cassetta, vi sveliamo tutti i segreti — pag. 2-3

□ **EXTENDED**
È un tema all'ordine del giorno: proteggere il software. Si può anche con l'Extended — pag. 4

□ **TI 99/4A**
Con il Clock, Bus ed Accessori, continuiamo il viaggio alla scoperta del nostro TI. Nella cassetta la 2ª lezione di «Teach Bit» — pag. 5-6

□ **EXTENDED + ESPANSIONE 32 K**
Se possedete sia Extended che Exp 32 K eccovi finalmente il modo di programmare in linguaggio macchina — pag. 7-9

□ **EXPANDER**
Un programma che carica in memoria un set di sotto programmi in linguaggio macchina. Gira in Extended — pag. 9-10

□ **PROGRAMMAZIONE**
Per la serie «Come costruire un arcade game» è la volta dello «Score Board» — pag. 11-12

□ **MINI MEMORY**
Parliamo di Standard e Extended utilites — pag. 13-15

□ **MERGE**
Dopo il MERGE su cassetta per l'Extended basic, è la volta del MERGE per la MINI MEMORY — pag. 16

□ **STATISTICA**
La correlazione di Rango.
Continuate a seguire questa rubrica, presto riuscirete a fare «13». — pag. 17-18

□ **IL CASTELLO NOTTE OSCURA**
Un classico Adventure sempre vario e coinvolgente. Buona grafica, gira in Basic — pag. 19-20

□ **LUNAR LANDER**
Un gioco simpatico in Extended — pag. 21

□ **ROCKFELLER**
Il simpatico corvo dal becco giallo è protagonista di questo gioco in Extended. Ohh...!! — pag. 22-23

□ **DIDATTICA**
Italiano. Un utile programma per chi vuole usare il TI 99 anche per studiare — pag. 24

□ **CALORIE**
Calcolate quante calorie «ingurgitate» in un pasto e regolate di conseguenza la vostra dieta — pag. 25

□ **MUSICAL AIDER**
Provate ad usare il TI 99 come tastiera musicale. È sorprendente! — pag. 26-27

□ **PICCOLI ANNUNCI** — pag. 28-29

□ **LA POSTA** — pag. 30

□ **COLLABORATE CON NOI** — pag. 31-32

□ **TI 99 MARKET** — pag. 33-34

TI 99 NEWSOFT

ANNO 2 - Numero sei
FEBBRAIO 1985
MENSILE
Direttore responsabile:
ANTONIO C. LOSITO
Coordinatore Editoriale:
VINICIO CIVININI
Direttore Amministrativo:
ENRICO CALLERIO
Art Director:
BEPPE RE-FRASCHINI
Illustrazioni:
MARIA TERESA COCOZZA
Segreteria:
EMANUELA CHIAPELLO
CRISTINA FRISON
Redazione:
FIORENZA AURIEMMA
DARIO BAREZZI
Collaboratori:
SERGIO BORSANI,
GIANPAOLO BOTTIN,
DANIELE CATALFAMO,
PAOLO CIVARDI,
STEFANO DELLI PONTI,
FRANCESCO MACALI,
MAURO MATTIAZZI,
EZIO MONTINI,
LEVIO PEROTTI,
RICCARDO ROSSI,
MARCO SQUINTANI,
ALBERTO STRAFILE,
SEBASTIANO TOMASELLO
TI 99 NEWSOFT è una pubblicazione della **NEWSOFT snc**, via Stefano Jacini 4 - 20121 MI
Tel. 02/807464 - Telex 324284
REGISTRAZIONE AL TRIBUNALE DI MILANO N. 323 DEL 7.7.84

© COPYRIGHT NEWSOFT snc
TUTTI I DIRITTI RISERVATI
Manoscritti e foto originali, anche se non pubblicati, non si restituiscono ed è vietata la riproduzione anche parziale di testi, fotografie e listati.

Pubblicità: AV STUDIO
Via Giuseppe Meda, 23
Tel. 02/8357155
Abbonamenti: l'abbonamento annuo a 11 numeri di *TI 99 NEWSOFT* costa L. 95.000 per l'Italia; L. 160.000 per l'Europa (spedizione aerea); L. 200.000 per tutti gli altri paesi. L'importo, per assegno non trasferibile o per vaglia postale, va inviato a NEWSOFT snc, via S. Jacini 4 - 20121 Milano.
Stampa: CGB, Albegno di Treviolo (BG)
Concessionaria per la distribuzione: EUROSTAMPA Srl - C.so Vittorio Emanuele 111 - 10123 Torino - Tel. 011/538166-7

L'editore non si assume responsabilità alcuna sul contenuto degli articoli e/o software inviati. Per gli articoli firmati e/o siglati da collaboratori esterni, la redazione si assume la responsabilità prevista dalle leggi sulla stampa.
Tutti possono collaborare a TI 99 NEWSOFT, sia con articoli che con programmi. Scriveteci o telefonateci in redazione per avere tutti i ragguagli.

downloaded by www.ti99iuc.it

GESTIONE FILES SU CASSETTA

Finalmente, dopo due mesi, siamo arrivati alla parte che più vi interesserà. Questa volta vi proponiamo delle semplici e pratiche routines di facile applicazione, che potranno trovare posto nei vostri programmi, senza creare alcun problema. Se ci avete seguito nelle precedenti puntate della nostra rubrica, non troverete alcuna difficoltà nel capire come sono state realizzate, e sarete in grado di poterle adattare nel migliore dei modi ai vostri programmi.
Le routines che vi proponiamo vi verranno presentate e commentate in ordine crescente di complessità.

Le variabili di matrice di riga 10, possono essere sostituite facilmente. Le abbiamo definite variabili di matrice perché esse hanno la caratteristica di essere in memoria e quindi di poter manipolare (cancelare, sostituire) i dati in esse contenuti.
Nell'inserimento dei dati è sempre consigliabile adottare sistemi di verifica dei dati introdotti, come nell'esempio alle righe 50, 60.
La routine di scrittura non comporta molte difficoltà, ma può essere migliorata ulteriormente, per ridurre il tempo di scrittura.
Sostituendo alle righe 110, 120, 130 le seguenti istruzioni, raddoppieremo il numero di dati che verrà salvato, dimezzando il tempo di scrittura.

```
110 FOR C=1 TO 10 STEP 2
120 PRINT #1 : A$ (C), B (C)
    A$ (C+1), B (C+1)
130 NEXT C
```

Se avessimo messo come passo di incremento un numero superiore a 2, avremmo dovuto aggiungere al passo 120 altre variabili, quanti erano i dati da inserire per ogni incremento. Per esempio se il passo di incremento fosse stato 3, il passo 110 avrebbe dovuto avere 3 gruppi di variabili:

```
110 FOR C = TO 10 STEP 3
120 PRINT #1:A$(C), B (C),
    A$ (C+1), B (C+1), A$
    (C+2), B (C+2)
130 NEXT C.
```

Questo continuo aggiungere ha un limite, che è posto da FIXED.
Se il valore del FIXED è posto a 64, noi potremmo aggiungere dati al record sino al raggiungimento di tale lunghezza, in questo caso di 64 caratteri.
Vi ricordiamo che il FIXED può avere un valore da 1 a 192.
Ovviamente, in fase di successiva rilettura dei dati, dovremo tenere conto del modo

Apertura e salvataggio dei dati contenuti in una matrice

```
10 DIM A$(10),B(10)
20 FOR S=1 TO 10
30 INPUT "NOME COGNOME CLIENTE ":A$(S)
40 INPUT "CODICE CLIENTE ":B(S)
50 INPUT "I DATI SONO ESATTI ?(S-N)":R$
60 IF R$="N" THEN 30
70 NEXT S
80 REM ********************
90 REM ROUTINE DI SCRITTURA
95 REM ********************
100 OPEN #1:"CS1",INTERNAL,OUTPUT,FIXED 64
110 FOR C=1 TO 10
120 PRINT #1:A$(C),B(C)
130 NEXT C
140 CLOSE #1
```

in cui sono stati registrati.
Se sono stati scritti più dati consecutivamente in un solo record, dovremo leggerne un ugual numero.
Solitamente si ricopia la routine di scrittura sostituendo la PRINT con una INPUT.
Dovremo anche aprire un altro file con le stesse caratteristiche, ma in modo INPUT.
Anche qui, possiamo ricopiare tranquillamente la OPEN sopra indicata, cambiando la scritta OUTPUT con INPUT.
Se nel nostro programma sono presenti più FILES con caratteristiche diverse, dovremo aprire tanti file quanti sono i gruppi con caratteristiche uguali.
Per esempio se in un programma dobbiamo scrivere e leggere dei dati, dovremo aprire due FILES, uno di scrittura e uno di lettura.
Il numero posto dopo il cancelletto «#», serve come riconoscimento, quindi non deve essere uguale a nessun altro FILE già aperto in precedenza.
La grandezza della matrice A$ può essere aumentata, ma ricordatevi di cambiare anche la lunghezza dei cicli FOR-NEXT delle righe 160, 240, 300
Non ci resta che dire: arrivederci alla prossima puntata

EZIO MONTINI

Esempio di doppia apertura di files input, output

```
70 DIM A$(10)
80  REM ******************
90  REM APERTURA DEI FILES
100 REM ******************
110 OPEN #1:"CS1",INTERNAL,INPUT,FIXED 64
120 OPEN #2:"CS1",INTERNAL,OUTPUT,FIXED 64
130 REM ******************
140 REM INSERIMENTO DATI
150 REM ******************
160 FOR S=1 TO 10
170 INPUT "NOME COGNOME CLIENTE ":A$(S)
180 INPUT "I DATI SONO ESATTI ?(S-N)":R$
190 IF R$="N" THEN 140
200 NEXT S
210 REM ******************
220 REM SCRITTURA DEI DATI
230 REM ******************
240 FOR A=1 TO 10 STEP 2
250 PRINT #2:A$(A),A$(A+1)
260 NEXT A
270 REM ******************
280 REM RILETTURA DEI DATI
290 REM ******************
300 FOR A=1 TO 10 STEP 2
310 INPUT #1:A$(A),A$(A+1)
320 NEXT A
```

Avete acquistato il Numero di Febbraio?

Ma sapete che sono usciti i Numeri di Giugno Settembre Ottobre, Novembre, Dicembre e Gennaio?

Per riceverli direttamente a casa vostra, scriveteci! Ve li invieremo in contrassegno a L. 10000 (+spese cadauno).

TI 99 NEWSOFT
Via S. Jacini, 4
20121 MILANO

PROTEGGERE IL SOFTWARE

Uno dei tanti problemi sorti negli ultimi anni è il cosidetto vampirismo di software. Purtroppo, il supporto magnetico costituito dalla cassetta è decisamente vulnerabile, qualunque sia la protezione del programma; basti pensare che ognuno di noi può duplicare fisicamente la cassetta con due registratori collegati in parallelo.
Questo metodo è il più diffuso, ed è impossibile evitarlo. Per nostra fortuna non tutti posseggono una doppia piastra per la duplicazione. Quindi, noi possiamo provare in vari modi ad impedire a quel malcapitato possessore del nostro programma, che ignaro delle tremende protezioni, provi il classico SAVE CS1 o DSK1, che dir si voglia, dopo averlo caricato.
In EXTENDED BASIC, esiste un modo semplice quanto sadico, per impedire non solo il list, ma anche il salvataggio e quindi il cambiamento del programma o la cancellazione di linee alla cieca, es. 1 «enter» ecc...
Per proteggere in tale modo i nostri programmi, basta salvarli in questo modo:

SAVA CS1, PROTECTED

Il programma salvato sulla cassetta, risulta protetto nella maniera sopra descritta per sempre.
ATTENZIONE: IL METODO DI PROTEZIONE È IRREVERSIBILE.
Assicuratevi di essere in possesso di una copia non protetta, e che il programma sia privo di errori.
Vi raccomandiamo di evitare di spedirci programmi protetti da pubblicare, perché detta protezione ci impedisce di valutare il vostro lavoro, ed ogni programma protetto che riceveremo sarà considerato nullo, quindi da cestinare. Questa protezione tuttavia non impedisce al possessore del programma di mandarlo in esecuzione.
Ci sono altri metodi di protezione meno drastici che possono impedire il list, o il run del programma in un certo modo.
Per esempio noi possiamo avvertire che il prgramma può essere mandato in esecuzione solo se caricato con una RUN «CS1», in caso contrario si fermerà causando il CRASH o blocco del sistema. costringendo il malcapitatc trasgressore a spegnere la macchina con conseguente perdita del programma.
Avrete notato che con il modulo EXTENDED BASIC, ci è consentito leggere il contenuto di alcune locazioni di memoria, diciamo alcune, perché ci sono delle locazioni che non sono leggibili da BASIC e la lettura accidentale o VOLONTARIA di tali locazioni, porta inevitabilmente al blocco del sistema.
La locazione decimale -26624, se letta da i risultati sopra citati.
La CALL PEEK (-26624, A), se eseguita blocca il sistema.
Il modo più semplice per fare capire al computer come è stato mandato in RUN un programma è di controllare l'angolo in basso a sinistra dello schermo.
Se il programma e mandato in esecuzione con una RUN «CS1», troveremo uno spazio vuoto, corrispondente al codice ASCII 32, letto con una CALL GCHAR.
In caso non positivo, passeremo il controllo del programma alla linea dove è situata la malefica CALL PEEK (-26624, A).
Ci dispiace concludere qui il discorso, ma non disperate, nel prossimo numero troverete i modi di protezione del listato e l'applicazione pratica di ciò che avete sino ad ora letto.

EZIO MONTINI

CLOCK, BUS ED "ACCESSORI"

IL CLOCK

Come abbiamo già accennato nel numero scorso, l'unità centrale di elaborazione o *CPU* deve poter conoscere quando porre fine ad una operazione e cominciare quindi quella seguente. Per ovviare a questo problema, all'interno della macchina è posto un dispositivo temporizzatore: il *clock*. Questo prezioso componente, come suggerisce il nome, compie un'opera di sincronizzazione delle operazioni svolte dalla *CPU*.
Il *clock* dunque emette impulsi elttrici cadenzati, fornendoli con una commutazione di tensione tra zero e cinque volt effettuata da un cirstallo di quarzo. Risulta ora chiaro come si possa, avendo la velocità del segnale, capire la velocità di lavoro della macchina dato che tanto più veloce sarà il susseguirsi degli impulsi tanto più celere sarà la reazione del microprocessore.

I BUS

Al pari di una metropoli ben organizzata anche i vari componenti dell'elettronica del nostro TI 99 sono messi in comunicazione tra loro per mezzo di *bus*, che nel nostro caso sono delle piste conduttrici di impulsi. Tali *bus* sono responsabili del trasferimento dei *bit*-informazione per le varie unità.
In un sistema a microprocessore i *bus* sono generalmente tre: uno relativo ai dati, l'altro per gli indirizzi e un terzo di controllo, che trasferisce le istruzioni dalla memoria alla *CPU*. I *bus* dati e di controllo codificano l'istruzione trattandola a gruppi di otto *bit*, quello degli indirizzi opera invece a sedici *bit*. Grande importanza hanno poi i *bus* di collegamento delle interfacce, dato che permettono di effettuare collegamenti di periferiche o espansioni senza dover manipolare le connessioni dei circuiti interni del calcolatore.
Interrompere una sola di queste piste comprometterebbe quindi l'intero funzionamento della macchina; poniamo per questo motivo una particolare attenzione sul loro operare, facendo un piccolo esempio di procedura che metta in risalto la sequenza degli eventi.
1) L'unità di controllo della *CPU* inizia un'operazione di lettura dalla memoria centrale di un dato, posto in una certa locazione. Come prima cosa invia l'indirizzo della locazione alla memoria sfruttando il *bus* indirizzi e, quasi simultaneamente, segnala con il *bus* di controllo che si tratta di un'operazione di lettura della memoria centrale e non magari di scrittura sulla memoria esterna (fig. 1).

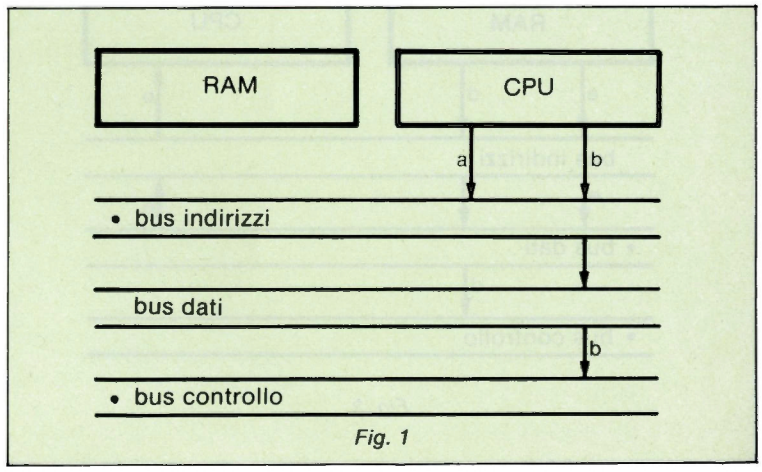


Fig. 1

2) La memoria centrale comincia a leggere il dato richiesto nella locazione che le è stata fornita segnalando all'unità di controllo di attendere. Per far questo, invia un'istruzione di messa in attesa lungo il *bus* di controllo, senza la quale la *CPU* continuerebbe tranquilla a eseguire le operazioni seguenti senza aver ricevuto il dato richiesto. (fig. 2)

3) Non appena la memoria ha letto il dato nella locazione segnalata, annulla il segnale di attesa e invia l'informazione grazie al *bus* dati. Finalmente l'unità di controllo ha ottenuto ciò che voleva ed è pronta a procedere all'istruzione sucessiva. (fig. 3)

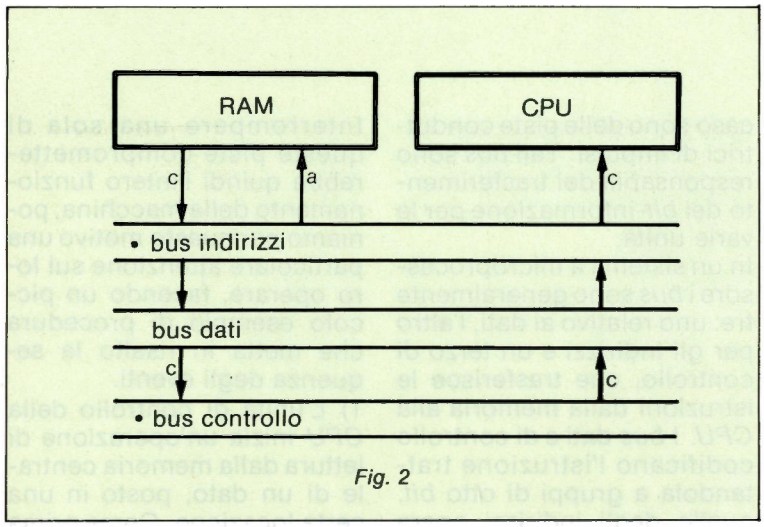


Fig. 2

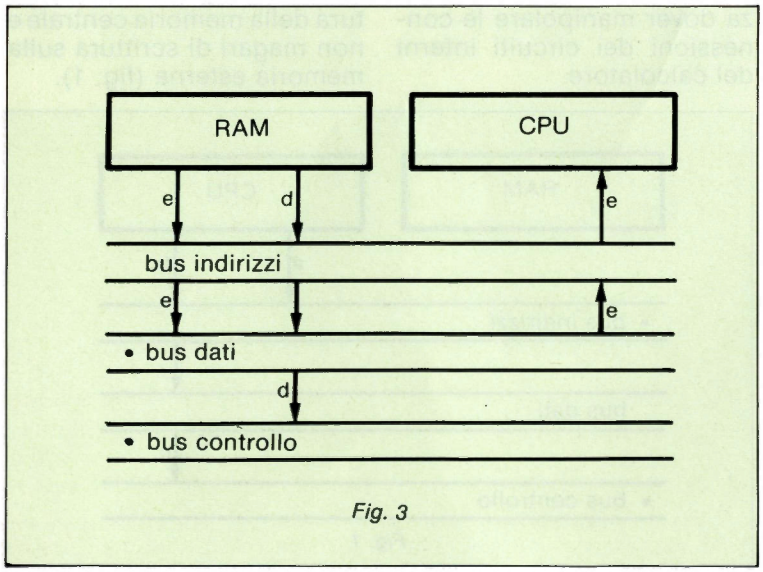


Fig. 3

ACCESSORI

Pur non potendosi definire parte integrante dell'*hardware* di un sistema, alcuni elementi svolgono importantissime funzioni che assicurano il corretto funzionamento della macchina. Uno di questi è, per esempio, l'alimentatore di tensione che, tanto criticato per la sua grossolana estetica a pedale, fornisce al computer la corrente elettrica alla tensione richiesta.
Date le sue grosse dimensioni, la Texas ha pensato bene di separarlo dall'unità centrale ponendolo in un contenitore di plastica nero e collegandolo con un primo cavo terminante in una presa a quattro poli (due superiori e due inferiori) da agganciare al computer e un secondo con la spina per la corrente secondo gli standard europei. È inoltre dotato di stabilizzatori per correggere gli eventuali sbalzi di tensione, che danneggierebbero i delicati microcomponenti elettronici del sistema, e di un fusibile da 250 V in caso di improvvise cadute di tensione (vedi black-out).
Poiché l'alimentatore resta sotto tensione anche se il computer è spento, in caso di assenza prolungata sarebbe opportuno staccare la presa di corrente o per i più bravi, inserire un interruttore tra la presa e il nostro "pedalone".

MARCO SQUINTANI

LINGUAGGIO MACCHINA

Eccoci qui come promesso, e per la gioia degli appassionati della velocità dei programmi, vi insegneremo come programmare in linguaggio macchina con l'ausilio dell'EXT. BASIC + EXP. 32K.
Programmare in linguaggio macchina significa scrivere dei programmi assembly direttamente con delle CALL LOAD.
La programmazione in linguaggio macchina presuppone una minima conoscenza dell'assembler del TMS 9900, che come sapete è il nostro processore.

Se ci avete fedelmente seguito nelle rubriche assembly e mini memory, non troverete difficoltà a capire cosa sono le word, i byte, bit, ecc.

Vi consigliamo di andare a rivedere questi concetti base, perché ci risulterebbe un po' difficoltoso ripeterli in questa rubrica.

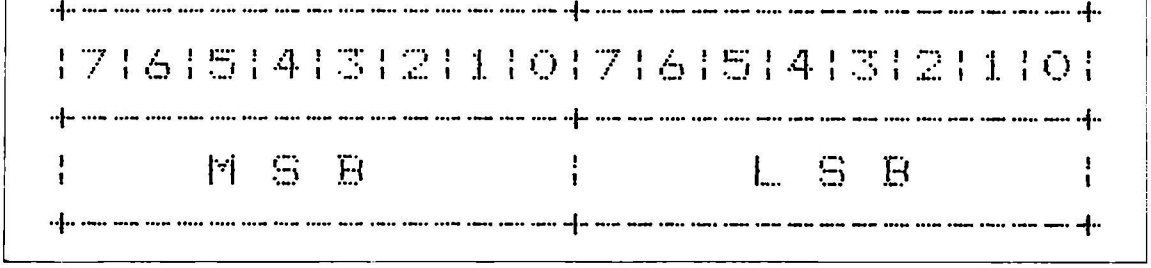
Il nostro processore è a 16 bit, ciò significa che lavora con delle word a 16 bit, a sua volta divisibili in due byte.

Il processore assegna ad ognuno di questi byte un'importanza.

Quello di sinistra è considerato come il MSB (Most Significant Byte), il più importante o significativo.

Quello si destra è considerato come LSB (Less Significant Byte), il meno importante o significativo.

Configurazione di word



Questa distinzione che apparentemente può risultare insignificante, assume molta importanza nel momento in cui noi dobbiamo tradurre un programma da assembly a linguaggio macchina.
Infatti nel MSB troviamo il codice di esecuzione dell'istruzione assembly e nel LSB troviamo il valore da noi inserito per eseguire quella determinata funzione.
Un esempio può essere facilmente dato da:
l'istruzione JMP ha come codice di esecuzione il valore esadecimale >10, detto valore sarà posto nel PSB; il LSB conterrà, espresso in esadecimale, il numero di locazioni da saltare, dato che JMP in assembly rappresenta un salto incondizionato di n. locazioni, si può paragonare al GOTO presente nel linguaggio BASIC.
Dato che il programma in LM deve essere inserito in memoria con delle CALL LOAD, e dato che le CALL LOAD non accettano valori esadecimali, dovremo prenderci la briga di tradurre pazientemente, istruzione per istruzione, tutto il programma assembly che vogliamo trasformare, tenendo conto che il MSB e il LSB, vanno tradotti separatamente in decimale.
Per esempio la word esadecimale >1003, dovrà essere tradotta >10 e >03, con conseguente traduzione in decimale 16,3. I valori così ottenuti dovranno essere inseriti in memoria consecutivamente ed in progressione di locazione.
In EXTENDED BASIC, possiamo cominciare ad introdurre i nostri programmi in LM dalla locazione decimale 12288, per evitare problemi. È tuttavia possibile cominciare ad introdurre dati da locazioni posteriori, cioè minori, a quella indicata, ma ciò richiede accurati calcoli per determinare la qualità di me-

moria che il computer richiede per il buon funzionamento del nostro programma in Linguaggio Macchina.

COME OTTENERE LA TRADUZIONE DA ISTRUZIONE ASS. A ESADECIMALE

Ci sono tre metodi per ottenere questa traduzione:
1) Possedere il modulo MINI MEMORY e ricopiare la traduzione scritta all'immediata sinistra dell'istruzione.
2) Essere in possesso del modulo ED. ASS. e fare un assemblaggio con codice oggetto e sorgente.
3) Conoscere l'assembly e tradurre a tavolino le istruzioni.
Il terzo modello presuppone che si sia in possesso del fatidico manuale dell'ED. ASSEMBLER, dove sono contenute tutte le traduzioni delle istruzioni.
Noi cercheremo di aiutarvi pubblicando delle tabelle di traduzione, istruzione per istruzione.
Vi ricordiamo che i programmi scritti per l'ED. ASS. o per MINI MEMORY, non sempre possono girare in ambiente EX. BASIC. Le differenze principali sono date dalla diversa locazione delle utility richiamabili con BLWP.
Alleghiamo una tabella di traduzione di utility da MINI MEMORY a EX. BASIC.
Una volta inserito il programma in LM, dovremo avvertire il computer che il nostro programma è pronto per essere eseguito.
Per poter fare ciò è necessario compiere due operazioni.
Con due CALL LOAD andremo ad inserire il nome del programma nella ref/def table e diremo al computer dove si trova l'inizio e la fine del programma.
Per prima cosa dovremo agire sulla locazione decimale 8194, introducendo due numeri che rappresentano la locazione di fine del programma.
Poi di seguito introdurremo due valori fissi che sono 63, 240.
I numeri di fine programma si ottengono dividendo la locazione finale per 256, poi moltiplicando il decimale per 256.
Se il risultato non ha decimale ed è quindi un numero intero, introdurremo come secondo numero uno zero.
Esempio: end loc. 12561, i numeri saranno:

TABELLA DI CONVERSIONE DA ED.ASS O MINI MEMORY A EXT.BASIC + 32K

BLWP nome utility	valore BLWP ed.ass,m.mem	valore BLWP ext.bas.+32K
XMLLNK	>601C	>2018
KSCAN	>6020	>201C
VSBW	>6024	>2020
VMBW	>6028	>2024
VSBR	>602C	>2028
VMBR	>6030	>202C
VWTR	>6034	>2030
NUMASG	>6040	>2008
NUMREF	>6044	>200C
STRASG	>6048	>2010
STRREF	>604C	>2014
ERR	>6050	>2034

12561/256=49.07, il primo numero è 49, poi O.O7*256=18, il secondo è 18.
La CALL LOAD sarà: CALL LOAD (8194, 49, 18, 63, 240)
Poi dovremo dare un nome al programma per farlo eseguire, chiamandolo con una CALL LINK.
Questa volta dovremo agire sulla locazione decimale 16368, qui dovremo inserire consecutivamente dapprima il nome del programma sotto forma di codice ascii, poi due numeri che rappresentano l'indirizzo della prima istruzione eseguibile in LM. Detti numeri si ottengono operando sulla locazione di memoria della prima istruzione eseguibile, con il metodo sopra indicato, seguendo le stesse regole.
Il nome del programma non deve superare i sei codici ascii 32 sino al raggiungimento di sei numeri, e questa regola va sempre osservata rigorosamente.
Esempio: il nome che vogliamo assegnare al programma è PROVA e la locazione dove si trova la prima istruzione eseguibile è 12288. La CALL LOAD che dovremo avere sarà: CALL LOAD (16368, 80, 82, 79, 86, 65, 32, 48, 0).
Per questa volta abbiamo concluso, nella cassetta troverete un programma di nome EXPANDER che tramite il linguaggio macchina fornisce sei nuove istruzioni al BASIC. Oltre che servirvi come esempio, vi sarà molto utile nella programmazione di giochi.

EZIO MONTINI

«EXPANDER»

Il programma di nome EXPANDER che trovate sulla cassetta, deve essere caricato con la formula OLD CS1, poi mandato in esecuzione con il classico RUN.
Questo programma carica in memoria un set di sottoprogrammi in linguaggio macchina, eseguibili con delle CALL LINK.
Una volta caricati in memoria, dopo che il RUN è stato eseguito, i sottoprogrammi rimangono memorizzati, anche dopo aver dato il NEW.
Gli unici modi per cancellarli è di eseguire una CALL INIT, un BYE, un QUIT e ovviamente di spegnere il computer.
I sottoprogrammi aggiunti sono sei, rispettivamente divisi in due gruppi. Il primo gruppo è costituito da tre sottoprogrammi, che hanno la funzione di memorizzare.

1) CALL LINK («SAVES»)

Questa istruzione permette di salvare un'intera pagina grafica, permettendone la successiva utilizzazione.
Come potrete costatare dalla sorgente del programma assembler, riportata in figura, il sottoprogramma in questione salva la mappa video, locata dalla locazione decimale 0 alla locazione decimale 768. Introducendone il contenuto in un apposito buffer chiamato SCR.

2) CALL LINK («SAVEP»)

Detto sottoprogramma salva la mappa dei pattern o caratteri, locata dalla locazione esadecimale>O3FO e prosegue per 912 BYTES, corrispondente a 8 bytes per carattere; i caratteri sono 114, equivalente a 30-143 (30 compreso).
Sono memorizzati in un buffer di 912 bytes, chiamato PAT.

3) CALL LINK («SAVEC»)

SAVEC permette di salvare l'intera mappa dei colori, locata a partire dalla locazione esadecimale >0800 e che prosegue per 32 bytes, equivalenti a 2 bytes per ogni gruppo di caratteri.
Sono memorizzati in un buffer di 32 bytes chiamato COL.
Il secondo gruppo è anch'esso costituito da tre sottoprogrammi aventi la specifica funzione di richiamare in memoria i dati salvati precedentemente negli appositi buffers.

1) CALL LINK («LOADS»)

Richiama dal buffer di nome SCR i dati della pagina grafica.

2) CALL LINK («LOADP»)

Richiama dal buffer di nome PAT i dati della mappa dei caratteri.

```
*****************************************
*                                       *
*   ESTENSIONE ISTRUZIONI EX.BASIC+32K  *
*                                       *
*   by EZIO MONTINI and MARCO CONTENTE  *
*                                       *
*              version 1.0              *
*                                       *
*   I sottoprogrammi sono richiamabili  *
*   direttamente da EX.BASIC+32K con    *
*   CALL LINK .                         *
*   Le CALL LINK possibili sono sei :   *
*                                       *
* CALL LINK("SAVES") CALL LINK("LOADS") *
* CALL LINK("SAVEP") CALL LINK("LOADP") *
* CALL LINK("SAVEC") CALL LINK("LOADC") *
*                                       *
*****************************************
```

```
        DEF SAVES,SAVEP,SAVEC
        DEF LOADS,LOADP,LOADC
SCR     BSS  768
PAT     BSS  912
COL     BSS  32
        EVEN
SAVES   MOV  R11,R10
        LI   R0,0
        LI   R1,SCR
        LI   R2,>02FF
        BLWP @>202C
        MOV  R10,R11
        B    *R11
SAVEP   MOV  R11,R10
        LI   R0,>03F0
        LI   R1,PAT
        LI   R2,912
        BLWP @>202C
        MOV  R10,R11
        B    *R11
SAVEC   MOV  R11,R10
        LI   R0,>0800
        LI   R1,COL
        LI   R2,32
        BLWP @>202C
        MOV  R10,R11
        B    *R11
LOADS   MOV  R11,R10
        LI   R0,0
        LI   R1,SCR
        LI   R2,>02FF
        BLWP @>2024
        MOV  R10,R11
        B    *R11
LOADP   MOV  R11,R10
        LI   R0,>03F0
        LI   R1,PAT
        LI   R2,912
        BLWP @>2024
        MOV  R10,R11
        B    *R11
LOADC   MOV  R11,R10
        LI   R0,>0800
        LI   R1,COL
        LI   R2,32
        BLWP @>2024
        MOV  R10,R11
        B    *R11
        END
```

3) CALL LINK («LOADC»)

Richiama dal buffer di nome COL i dati di mappa dei colori.

Vi accorgerete come questo semplice programmino in LM migliorerà i vostri giochi, permettendo immediati cambi di schermo, pattern, colori, con il solo comando CALL LINK, specificando il nome della utility da voi desiderata.
Le CALL LINK possono essere eseguite sia in modo comando che in modo esecutivo. Il programma in EXT. BASIC, contenuto nella cassetta, potrà facilmente trovare posto nei vostri programmi date le sue ridotte dimensioni e il suo scarso utilizzo di memoria (Kb utilizzati=1.230).
Alleghiamo un piccolo programmino di esempio per mostrarvi l'enorme potenzialità del nostro programma.
Prima di eseguire il programma di esempio, assicurarsi che il programma EXPANDER sia già stato caricato in memoria.

EZIO MONTINI

COME COSTRUIRE UN ARCADE GAME: GLI SCORE BOARD

Ogni volta che giochiamo con un videogame vorremmo riuscire a battere il record del gioco e non a caso si fanno gare, si cerca di battere record di durata e resistenza, insomma ci piace la competizione, tutto questo è reso possibile, dal gioco ed ovviamente da il contapunti, per intenderci lo «score».

Se qualcuno ricorda, i primi giochi si limitavano a segnare i punti fatti nella partita, solo in seguito cominciavano a tenere memorizzato anche il punteggio più alto realizzato e poi il massimo lo «Score board» cioè il tabellone che richiamava i migliori 10 punteggi con il nome o le iniziali di chi lo realizzava.

Dopo questa breve storia, passiamo alla nostra solita routine. Infatti questa puntata è dedicata alla creazione di uno «Score board», ed alla visualizzazione dei punteggi.
Anche per i punti è necessario preparare una ruotine a parte mentre il richiamo della sudetta dovrà avvenire solitamente dalla ruotine di Distruzione Nemico, che vedremo in una delle prossime puntate, dove viene dato il consenso ad incrementare il settore punti.

Ovviamente in TI BASIC, non disponendo dell'istruzione «DISPLAY AT», è necessario creare anche una routine di aggiornamento punti che permetta la stampa del punteggio al suo posto senza effettuare lo scorrimento dello sfondo.
Quando preparerete il quadrato di gioco sarà opportuno creare una stringa che servirà per stampare i punti, che potrebbe essere impostata in questo modo:

```
370 HPUNTI$="0000"
380 PUNTI$="0000"
390 P$="SCORE:" & PUNTI$
400 R=1
410 C=3
420 GOSUB 5000
430 P$="HI-SCORE:"
    &H PUNTI$
440 R=1
450 C=18
460 GOSUB 5000
```

Queste poche righe servono per creare la scritta azzerata del contatore punti in cima al quadro di gioco, i vettori "R" e "C", sono le coordinate di partenza delle scritte mentre alla riga 5000, dove ci manda la "GOSUB", c'è la ruotine che chiameremo di "PRINT AT".

Ed ora il listato "PRINT AT":

```
5000  REM PRINT AT
5010  Z=LEN(P$)
5020  FOR CF = 1 TO Z
5030  CALL HCHAR(R,C+CF,
      ASC(SEG$(P$,CF, 1)))
5040  NEXT CF
5050  RETURN
```

Ovviamente questa routine può servirvi anche ad altri scopi. Basta dare un valore alla «RIGA» ed alla «COLONNA», definire la Stringa, ed essa verrà stampata a partire dalle posizioni da voi definite.
Una volta analizzato il gioco, ovviamente bisognerà creare una tabella punti affinché il computer assegni ad ogni nemico abbattuto il punteggio corrispondente.
I metodi sono molteplici, noi ve ne elenchiamo uno dei tanti:

```
Creazione matrice
contenente i punteggi

  10  DIM A (4)
  20  FOR C=1 TO 4
  30  A(C)=C*100
  40  NEXT C

3000REM NEMICO COLPITO
```

GIOCHIAMO CON IL TI 99/4A

Un libro che non deve assolutamente mancare nella vostra libreria.

Potrete riceverlo a casa vostra a sole **L. 9500** + Spese di spedizione.

Per ogni richiesta di contrassegno, utilizzate il tagliando qui sotto spedendolo a:

NEWSOFT snc
via S. Jacini, 4
20121 MILANO

Vogliate spedirmi in contrassegno di L. 9500 + spese il libro: «GIOCHIAMO CON IL TI 99/4A»

Cognome .....

Nome .....

Indirizzo .....

downloaded by www.ti99iuc.it

Cap. ..... Loc. .....


```
3090 AS=X — Valore ASCII carattere nemico colpito
3100 AS=AS-127 — Sottrarre il numero ASCII dell'ultimo
3105 REM ROUTINE PUNTI — carattere prima di quelli nemici
3110 PUNTI=PUNTI+A(AS)
3120 P$=STR$(PUNTI) — Determinazione COLONNA DI PARTENZA STAMPA
3140 GOSUB 5000
3150 P$=""
3160 C=O
3170 RETURN
```

Questa routine quindi permette di stampare i punti al posto prestabilito, ovviamente i nemici, o meglio, i caratteri definiti, dovranno essere consecutivi e cominciare dal carattere 128 in modo che (in questo esempio sono stati considerati 4 nemici diversi), quando viene effettuata la sottrazione alla linea 3100 i numeri possibili sono dall'1 al 4, coprendo così l'intera matrice contenente i punti.
Se i punteggi invece non sono differenziati, ma uguali per tutti i nemici si possono evitare tanti intoppi eliminando le righe da 10 a 40 e le righe da 3090 a 3100 ed inserendo al posto della riga 3110 la seguente riga:

```
3110  PUNTI=PUNTI+100
```

Qui direi di sospendere la chiacchierata sul calcolo dei punti, anche perché il discorso molto generico si presta a tante varianti, tuttavia ritengo che la ruotine che vi abbiamo presentato possa essere interessante e anche piuttosto veloce.
Ora invece per i perfezionisti vedremo di presentare una routine di memorizzazione dei record e delle iniziali, fedeli alle tradizioni delle più famose Software-House (la trovate nella cassetta).
Anche qui il discorso si presta a tante variazioni sul tema e perciò abbiamo pensato di rivolgere la nostra attenzione al tipo più usato, cioè a quello che elenca 5 punteggi e quindi con il movimento del Joystick (nel nostro caso anche da tastiera) o a destra o a sinistra, c'è lo scorrimento in un riquadro del Set Alfabetico, e quando la lettera interessata appare, basta premere Enter o il tasto "FIRE" e la lettera appare al proprio posto.

MMG
MAURO MATTIAZZI

STANDARD E EXTENDED UTILITIES

Nel numero di gennaio abbiamo cominciato a parlare della ROOM del nostro modulo: abbiamo preso in considerazione i vettori e la NAME LINK ROUTINE. Si è fatto, in questo modo, qualche passo in avanti nella comprensione del funzionamento «interno» del computer (a livello software). Questo mese ci occuperemo in dettaglio delle più importanti tra le extended e standard utilities, necessarie per la programmazione in Assembler.

GPLLNK: consente al programmatore di utilizzare le routines in GPL, residenti nella GROM. Possiamo così inizializzare il sistema, caricare i diversi sets di caratteri, eseguire operazioni matematiche, trigonometriche, etc.

Per fare ciò è sufficiente dare un «BLWP ∂>6018» (che è l'indirizzo del vettore della GPLLNK), seguito da un DATA con un numero che identifica la GPL routine da far eseguire.

Un esempio:

```
BLWP ∂>6018
DATA >0036
```

Il computer esegue un suono di warning (il solito).

All'indirizzo >837C troviamo il GPL STATUS BYTE: esso fa da STATUS REGISTER mentre sono in esecuzione le routines in GPL. Il bit 2 (condition bit) è settato dalla DSR routine per indicare che un file non esiste.

Ricodiamo che è opportuno azzerare lo STATUS BYTE prima di chiamare una delle GPL routines per evitare che ci venga segnalato un errore privo di significato.

Nella locazione >8354 è indicato il tipo di errore che eventualmente capita durante l'esecuzione delle floating-point routines, secondo un codice descritto a pagina 42 del manuale MM.

Fra le varie routines richiamabili, segnaliamo la CNS, identificata dal codice >14: essa ci permette di scrivere numeri sullo schermo poiché li converte in stringhe ASCTI. Il numero da convertire deve essere memorizzato in FAC nel formato floating-point. Spendiamo due parole riguardo a questo formato: i numeri floating-point (in italiano: «in virgola mobile»), che noi usiamo abitualmente programmando in linguaggi evoluti come il BASIC, non sono direttamente «masticabili» dalla CPU, che lavora con gli integers (numeri interi da 8 o 16 bits — > FF o>FFFF).

È questo il motivo per cui troviamo nella ROM delle routines che ci permettono operazioni con i numeri in virgola mobile. Il numero è contenuto in 8 bytes, secondo la notazione «radix 100» (A*100ΔB): il primo byte indica l'esponente, incrementato di >40, a cui elevare 100 (è quindi B+>40). Ciò significa che se ha un valore di >43 l'esponente è 3; con un valore di >3A è -5.

Per indicare che un numero è negativo i primi due bytes sono espressi nella forma two-'s-complement.

Gli altri sette bytes contengono A convertito nella notazione esadecimale; in particolare il secondo byte rappresenta il valore della parte del numero espresso in «radix 100 notation».

XMLLNK: essa ci permette il linking con le ROM-routines. Ci sono due modi per indicare quale routine si vuole far eseguire: nella word (che chiameremo word A) di seguito all'istruzione «BLWP ∂>601C» si può memorizzare, tramite lo statement DATA, il codice d'indentificazio-

ne (ID code) oppure l'indirizzo di partenza della routine.
Parliamo del codice: con esso possiamo riferirci nominalmente ad una delle 16 tabelle presenti in memoria, ognuna delle quali contiene 16 indirizzi che rappresentano gli entry points delle routines. Il codice viene memorizzato nel byte più significativo della word A; il nybble alto indica la tabella a cui ci si vuole riferire. Il secondo nybble indica la word che interessa nella tabella specificata.
Solo 8 (da >0 a >7) sono le tabelle utilizzabili, poiché quando è settato il bit 0 della word A, allora il contenuto è considerato un indirizzo: è questo il secondo metodo di linking. Nella figura 1 sono indicati gli indirizzi delle tabelle.
Facciamo notare che la CIF (con codice >23) è inutilizzabile: il manuale Editor-Assembler ci avverte che essa è caricata nell'espansione di memoria dell'E-A Loader (la tabella 2 è all'indirizzo >2000).
Tuttavia è stata inserita nella ROM del modulo all'indirizzo >618C; per utilizzarla è sufficiente digitare queste istruzioni:

```
BLWP ∂>601C
DATA >7200
```

KSCAN: questa routine chiama a sua volta la SCAN nella consolle, situata all'indirizzo >000E.
Un accesso diretto alla SCAN è possibile, ma bisogna ricordarsi di caricare (load) il GPL Workspace (operazione svolta dalla KSCAN). Questa effettua una scansione della tastiera (come il CALL KEY del BASIC), rilevando il tasto premuto e aggiornando lo STATUS (sempre >837C).
Nella locazione >8374 viene specificata l'unità di tastiera; in >8375 viene memorizzato il codice ASCII del tasto premuto (>FF se nessun tasto è stato premuto); in >8376->8377 è indicata la posizione dei joystick nel solito modo (>04, >00, >FC). Il bit 2 (e non 5) dello STATUS è settato quando è stato premuto un tasto diverso dal tasto premuto durante l'ultima scansione con la routine. Nella fig. 2 vedete un esempio di utilizzazione della routine, confrontato con un frammento di programma in BASIC. Bisogna ricordarsi di azzerare sempre lo STATUS BYTE: se, ad esempio, tralasciate l'istruzione «MOVD RO, ∂>837C»el programmino, il computer vi segnalerà «ERROR CODE 00».
VSBW, VMBW, VSBR e VMBR: sono quattro utilissime routines che ci facilitano il compito di scrivere e leggere nelle locazioni della VDP RAM (e quindi sullo schermo). Nella fig. 3 trovate i disassemblati di queste, con relative spiegazioni.
Come potete vedere, esse utilizzano un'unica subroutine (>616C) che scrive l'indirizzo di scrittura o di lettura nell'apposita locazione (>8C02-VDPWA). Ricordiamo che in essa non è possibile passare un word-address: è necessario muovere dapprima il byte meno significativo e successivamente (deve passare qualche microsecondo) il più significatio.
Le routines di lettura utilizzano la subroutine a partire da >6174: le prime due istruzio-

Numero tabella	Indirizzo
>0	Specificato in > OCFA
>1	Specificato in >OCFC
>2	> 2000
>3	> 3FC0
>4	> 3FE0
>5	> 4010
>6	> 4030
>7	> 6010

Figura 1 - Gli indirizzi delle tabelle.

```
10 CALL KEY (O, R, S)      CLR RO
20 IF R<>83 THEN 10        MOVB RO, ∂>8374     Unità tastiera: zero
30 END                     A BLWP ∂>6020       Scansione
                           MOVB ∂>8375,1       Lettura Codice ASCII
                           CI R1, >5300
                           JNE A
                           MOVB RO, ∂>837C     Pulisce lo Status
                                               Byte
                           B *R11              Ritorna
```

Figura 2 - Esempio KSCAN.

```
■ VSBW
>6126   BL ∂616C
        MOV ∂>0002 (R13),      Scrive un singolo byte
        ∂>8C00                 (>8C00 è il VDPWD)
        RTWP                   R13 dopo un BLWP contiene
                               il WP. ∂>0002 (R13) rappresenta
                               il vecchio R1.
■ VMBW
>6132   BL ∂>616C
>6136   MOV *R1 +, ∂>8C00      Ciclo di scrittura. Il VDP
                               RAM WRITE ADDRESS è
                               autoincrementante
        DEC R2
        JNE >6136
        RTWP
■ VSBR
>6040   BL ∂>6172
        MOVB ∂>8800, 0002(R13) Legge e muove in R1
        RTWP                   (>8800 è il Registro di
                               Lettura).
■ VMBR
>614C   BL ∂>6172
>6150   MOVB ∂>8800, *R1+      Ciclo di Lettura.
                               Anche il Registro di Lettura
                               è autoincrementante
        DEC R2
        JNE >6150
        RTWP
■ SUBROUTINE
>616C   LI R1, >4000           Per il set bit 1 con SOC 1, 2
                               routine di scrittura.
        JMP >6174
        CLR R1
>6174   MOV *R13, R2          Vecchio RO in R2
        MOVB ∂>7097, ∂>8C02    Scrive il byte meno significativo
                               di R2 (vecchio RO) nel registro
                               di scrittura indirizzo.
        SOC R1, R2
        MOVB R2, ∂>8C02
        MOV ∂>0002 (R13), R1   Per le routine con multide
                               bytes.
        MOV ∂>0004 (R13), R2
        B *R11                 Ritorno.
■ VWTR
>615A   MOV *R13, 1            Vecchio R0 in R1.
        MOVB ∂>0001 (R13),     Passaggio valore.
        ∂>8C02
        ORI R1, >8000          Set registro (bit 0).
        MOVB R1, ∂>8C02        Passaggio Registro.
        RTWP
```

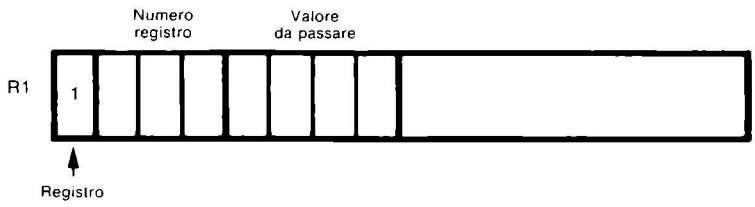


Figura 3

ni sono destinate alle routines di scrittura. Infatti il bit 1 del most-significant byte dell'indirizzo deve essere settato per indicare un'operazione di scrittura e ciò è eseguito dalle istruzioni.

```
LI RI,>4000
SOC RI,R2
```

Ripetiamo ora qualcosa che è indispensabile conoscere per riuscire a scrivere sullo schermo con un programma in LM chiamato da BASIC. Per fare questo NON bisogna utilizzare i soliti codici ASCII, ma si deve aggiungere >60 o 90 ad essi (lo si scopre a pag. 61 del manuale MM dove si legge: «...from BASIC there is a SCREEN BIAS of >60. SCREEN VAL=ASCII VAL+>60). Nel manuale E-A si parla di «SCREEN OFFSET of >60».
L'ultima utilities di cui ci occuperemo in questo numero è la VWTR. Ci serve per cambiare i valori degli otto VDP WRITE-ONLY REGISTERS. Dal disassemblato si può vedere che non appena è stato «spedito» il byte meno significativo (che contiene il valore al quale si vuole portare il registro specificato dal primo byte), il bit 0 di quest'ultimo è settato per indicare che non viene spostato un indirizzo. Con questo vi salutiamo: il prossimo mese continueremo la rassegna.

STEFANO DELLI PONTI

ABBONATEVI
A TI 99
NEWSOFT.
CON SOLE
L. 95.000
RICEVERETE
11 NUMERI
DELLA RIVISTA
E IN PIU'
AVRETE
DIRITTO AGLI
SCONTI SUL
SOFTWARE
E
L'ARDWARE
OFFERTO DA
TI 99 MARKET

Per abbonarsi
basta inviare
un assegno
non trasferibile
o un vaglia
postale a:

NEWSOFT SNC
Via S. Jacini, 4
20121 MILANO

MERGE PER MINI MEMORY

Nel numero di Dicembre abbiamo pubblicato il merge per extended basic ripromettendoci di pubblicare la versione per mini memory in questo numero.
Questo merge al contrario di quello per extended, fonde direttamente il programma in memoria con un programma memorizzato su nastro. Il merge (in effetti il termine "merge" implica anche un operazione di sort numeri di linea, cosa che questo programma non effettua) accoda il programma in memoria ad un programma su nastro nel momento in cui questo viene caricato in memoria.
Questo significa che prima del merge avremo sulla cassetta la prima parte del programma, mentre in memoria avremo la seconda parte. Per effettuare il merge dovremo battere:
CALL LINK («MERGE»)
eseguire le istruzioni sul video.
Facciamo un esempio:
Abbiamo due programmi
10 PRINT «PRIMO PROGRAMMA»
e
20 PRINT «SECONDO PROGRAMMA»
La sequenza per unirli è la seguente:
1) caricate, se non lo avete già fatto, il merge con il comando «L» dell'easy bug
2) torniamo in basic e carichiamo il secondo programma
20 PRINT «SECONDO PROGRAMMA»
3) battiamo CALL LINK («MERGE»), sul video compariranno le istruzioni per il registratore. Carichiamo a questo punto il programma che si trova sul nastro, quando il cursore ricomparirà sul video il merge tra due programmi sarà stato effettuato. Se non ci credete provate a listare il programma.
Contrariamente all'altro programma non esistono particolari limitazioni riguardo la lunghezza dei singoli programmi, comunque bisogna tener presente che la somma della lunghezza dei due programmi, più la lunghezza della tavola dei numeri di linea del secondo programma non deve superare i 14 Kb di memoria, in caso contrario il comando viene abortito.
Con questo abbiamo momentaneamente colmato uno dei buchi più grossi nel basic del TI. Se avete altri problemi non esitate a scriverci, nel limite del possibile vedremo di realizzare le vostre richieste.

RICCARDO ROSSI

CORRELAZIONE DI RANGO

Nella puntata precedente abbiamo pubblicato un programma che ci permetteva, avendo a disposizione dati numerici, di trovare la correlazione esistente tra due serie di dati X e Y.
Qualora non si disponga di precisi dati campionari, per esempio nel caso di valutazioni svolte da giudici diversi chiamati a dare valutazioni soggettive su una prestazione sportiva, su un concorso o, semplicemente, nel caso debbano decidere quale bevanda abbia un gusto migliore, quale detersivo sia più efficace, ecc. sarà necessario ricorrere al calcolo del coefficiente di Spearman.

Nota per i sistemisti al totocalcio:

Avrete notato che il lunedì, dopo la partita, i giudizi sui singoli giocatori delle squadre variano enormemente da giornale a giornale. Applicando il programma che stiamo per mostrarvi avremo modo di assegnare pesi diversi ai giudizi dei giornalisti, farci un'idea più chiara circa lo stato di forma di ciascuna squadra e, magari, scartare valutazioni perennemente difettose o troppo iperboliche.

Il coefficiente di Spearman

Per iniziare i lettori alla comprensione del programma possiamo cominciare con un esempio.
Due giornalisti sportivi hanno valutato le 16 squadre di serie A.

Sommando i valori dei voti dati ai giocatori di ciascuna squadra scopriamo che le squadre, chiamiamole A, B, C, D, E, F, G, H, I, L, M, N, O, P, Q, R per semplicità, sono state classificate in modo diverso dai due giornali.

Forniamo una tabella semplificativa

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1° giornale	B	F	C	A	E	D	G	N	L	I	R	P	Q	O	M	H
2° giornale	C	B	F	G	E	A	D	Q	R	M	L	P	N	I	O	H

Considerazioni

A prescindere dal fatto che le valutazioni sono manifestamente diverse, e questo già lo pensavamo, il pallone è rotondo, e variabile è anche il modo di interpretare il calcio, appare chiaro che, nel caso in cui nessuna squadra fosse stata classificata nella stessa posizione, allora i giornalisti non si sarebbero trovati d'accordo neanche a Natale, come a volte succede, e per noi il discorso statistico si sarebbe concluso.
Abbiamo invece, in tre casi, gli stessi piazzamenti, ciò non significa che di fronte a giudizi così unanimi la squadra H, per esempio, sarà quella che, concordemente condannata, finirà diritta in serie B, ma che, almeno, è quella che manifesta, per il momento, il grado di forma peggiore, e di ciò potremmo tenere qualche conto statistico.

Ho detto potremmo in quanto prima è, come al solito, indispensabile verificare che le coincidenze non siano avvenute per caso, poiché, allora ritorneremmo al discorsc precedente del Natale: lasciamo perdere, che è meglio!

Se non voltate pagina, ora controlliamo.
(Per i non portati alla matematica: saltate pure qualche riga ora, tanto poi si capisce lo stesso).
Il coefficiente di Spearman rho è calcolato dalla formula:

$$\rho = 1 - \frac{6 \Sigma D^2}{n (n^2 - 1)}$$

dove

D è la differenza tra ranghi per ogni elemento
Σ è la somma dei quadrati di D
n è il numero dei soggetti

rho può variare, come al solito, da +1 a —1; intendendo con +1 una relazione diretta, con 0 la mancanza di qualsiasi correlazione, e con —1 una relazione inversa. La tabella in alto illustrerà meglio il metodo di calcolo.

Applichiamo la formula risolutiva di rho:

$$\rho = 1 - \frac{6 \times 120}{16 (16^2 - 1)} = 0.8236$$

Questo valore 0.8236 ci è necessario per stabilire la possibilità che una siffatta correlazione sia avvenuta per caso.
Dall'osservazione della tabella allegata notiamo che per 16 campioni, la possibilità al 2.5% è uguale a 0.5000. Poiché 0.8236 è maggiore di 0.5000 possiamo affermare che esistono meno del 2.5% di possibilità che una correlazione così buona, o migliore, sia avvenuta per caso.
Possiamo pertanto ritenere che i voti siano stati assegnati con competenza, e con una certa uniformità di giudizio, e questo era quanto ci necessitava stabilire.

PAOLO CIVARDI

squadra	*giornale 1*	*giornale 2*	*differenza di rango D*	D^2
A	4	6	—2	4
B	1	2	—1	1
C	3	1	2	4
D	6	7	—1	1
E	5	5	0	0
F	2	3	—1	1
G	7	4	3	9
H	16	16	0	0
I	10	14	—4	16
L	9	11	—2	4
M	15	10	5	25
N	8	13	—5	25
O	14	15	—1	1
P	12	12	0	0
Q	13	8	5	25
R	11	9	2	4
		Totali	0	120

Probabilità che rho avvenga per caso

	20%	10%	5%	2%	1%	0.2%
4	.8000	.8000				
5	.7000	.8000	.9000	.9000		
6	.6000	.7714	.8286	.8857	.9429	
7	.5357	.6786	.7450	.8571	.8929	.9643
8	.5000	.6190	.7143	.8095	.8571	.9286
9	.4667	.5833	.6833	.7667	.8167	.9000
10	.4424	.5515	.6364	.7333	.7818	.8667
11	.4182	.5273	.6091	.7000	.7455	.8364
12	.3986	.4965	.5804	.6713	.7273	.8182
13	.3791	.4780	.5549	.6429	.6978	.7912
14	.3626	.4593	.5341	.6220	.6747	.7670
15	.3500	.4429	.5179	.6000	.6536	.7464
16	.3382	.4265	.5000	.5824	.6324	.7265
17	.3260	.4118	.4853	.5637	.6152	.7083
18	.3148	.3994	.4716	.5480	.5975	.6904
19	.3070	.3895	.4579	.5333	.5825	.6737
20	.2977	.3789	.4451	.5203	.5684	.6586
21	.2909	.3688	.4351	.5078	.5545	.6455
22	.2829	.3597	.4241	.4963	.5426	.6318
23	.2767	.3518	.4150	.4852	.5306	.6186
24	.2704	.3435	.4061	.4748	.5200	.6070
25	.2646	.3362	.3977	.4654	.5100	.5962
26	.2588	.3299	.3894	.4564	.5002	.5856
27	.2540	.3236	.3822	.4481	.4915	.5757
28	.2490	.3175	.3749	.4401	.4828	.5660
29	.2443	.3113	.3685	.4320	.4744	.5567
30	.2400	.3059	.3620	.4251	.4665	.5479

IL CASTELLO NOTTE SCURA

Il Castello notte scura è un gioco totalmente nuovo che si avvicina allo spirito delle «avventure» pur risultando sempre vario e coinvolgente ogni volta che viene giocato.

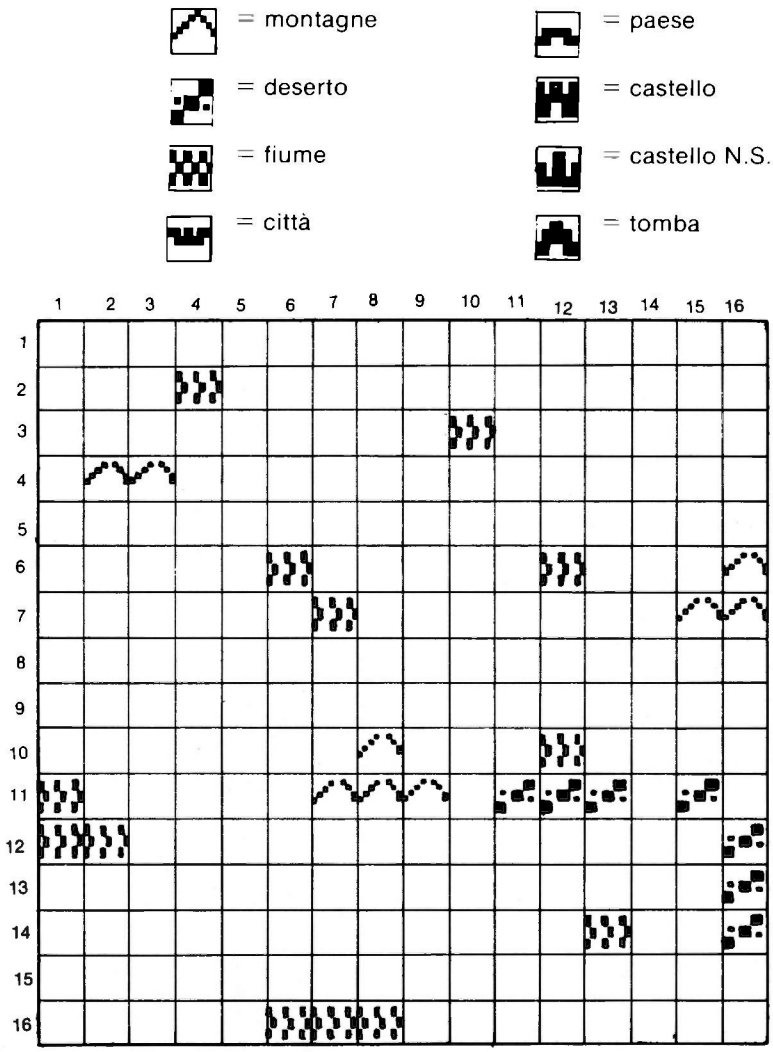
Lo scopo del gioco è triplo: inizialmente bisogna cercare di accrescere il numero e la forza dei soldati della propria armata. Poi bisogna trovare 5 chiavi che sono necessarie per entrare nel castello notte scura. Infine, una volta raccolte le chiavi, bisogna entrare nel castello e sconfiggere i suoi occupanti.

Le chiavi si possono trovare nei castelli o nelle tombe e si possono conquistare, insieme ad un cospicuo tesoro, dopo aver ucciso tutto i loro difensori.

In seguito ad ogni combattimento il luogo visitato sarà nuovamente riempito di mostri e di tesori.

I castelli e le tombe, insieme ad altri luoghi vari ed interessanti, sono sparsi in un territorio di 16x16 unità di lato.

Ogni unità è una possibile locazione che potete occupare. Sul video viene mostrata una sezione di 9x9 unità con la vostra armata sempre al centro. Ci sono 8 caratteri grafici speciali che sono:



Questa è la mappa che contiene le particolarità geografiche del campo di gioco e rimane invariata per tutte le partite. Si dovranno poi aggiungere le città, i castelli e le tombe che cambiano posizione ad ogni partita.

Nelle città e nei paesi (i cui mercati sono più cari di quelli cittadini), potete acquistare tutto ciò che serve allo svolgimento del gioco: armature ed armi, che accrescono la forza dei vostri uomini; informazioni sulle locazioni delle chiavi; cibo, che serve per nutrire l'armata ad ogni spostamento (l'acqua si trova solo nei fiumi); uomini, divisi in soldati e ladri (questi ultimi sono più cari ma permettono di derubare i castelli ed impossessarsi dei loro tesori senza affrontare i loro difensori). Quando acquistate cibo o uomini dovete indicare le quantità sempre con un numero di 2 cifre (es. per comprare 4 soldati dovrete premere 0/4).

Tabella riassuntiva degli acquisti possibili nei mercati

(i prezzi sono riferiti ai mercati cittadini)

1) CIBO: prezzo = 3 pezzi d'oro per unità per persona
2) INFORMAZIONI: prezzo = da 1500 a 3500 pezzi d'oro
3) UOMINI: soldati prezzo = 550 + equipaggiamento - ladri prezzo = 675 + equipaggiamento.

Armi e armature

	Prezzo	Fatica	Forza
Leggera	50	2	1
Media	200	5	3
Pesante	400	10	5
Daga	45	1	2
Ascia	200	2	5
Spada	350	5	7

Per calcolare la forza dei vostri soldati dovete sommare la forza dell'armatura + 1 (es. all'inizio con l'armatura leggera e la daga, la vostra forza sarà 1+2+1=4).
Il coeff. fatica influenza invece la fatica che farete spostandovi e si trova sommando la fatica delle armi e delle armature +15.

All'inizio del gioco viene però posto convenzionalmente a 15. Ad ogni combattimento e ad ogni spostamento aumenta la fatica che, se raggiunge il 100%, può avere effetti letali sui vostri soldati. Pozioni e/o turni di riposo azzerano o diminuiscono l'affaticamento dell'armata.

Quadro riassuntivo delle caratteristiche dei nemici

(tra parentesi la loro forza di combattimento)

	Aperto	Montagne	Deserti	Fiumi
Portatori tesori	Giganti (8)	Gnomi (5)	Predoni (11)	Tritoni (7)
Portatori pozioni	Vampiri (5)	Yeti (8)	Uomini sabbia (8)	Pirati (10)

NEI CASTELLI: SOLDATI (15)
NELLE TOMBE: SPETTRI (12)
NEL CASTELLO NOTTE SCURA: DRAGHI (forza variabile).

Oltre che nei castelli e nelle tombe i nemici vi possono attaccare anche allo scoperto. Vi sono differenti tipi di nemici che si possono dividere in 2 classi: 1) portatori pozioni 2) portatori tesori ed eventualmente chiavi per il castello nottescura. Entrambe le classi possono portarvi indubbi vantaggi ma bisogna attaccare solo se si è in superiorità numerica per non rischiare di essere sconfitti.
La durata media di una partita (se riuscite a non farvi eliminare al primo combattimento), varia dalle 2 alle 4 ore e sono sicuro che anche voi, dopo i primi disastrosi tentativi, potrete entrare nel terribile "CASTELLO NOTTE SCURA".

LUCA MUZIO

LUNAR LANDER

Dopo tanti utility eccoci a voi con un gioco veramente notevole, sia per la grafica che per la dinamica.
È fuori dubbio che il gioco prende spunto dalla cartuccia Commodore dal titolo "Jupiter Lander", tuttavia ci sono alcune considerazioni da fare:
la prima è che Lunar Lander è eseguito tutto in Extended Basic, mentre l'originale è stato eseguito in linguaggio macchina, e che per ovvi motivi, di linguaggio appunto, il programma non offre la ZOOMATA, cioè quando siete in prossimità di una coincidenza, l'originale ingrandiva il particolare in avvicinamento.
Tuttavia possiamo dire che questo gioco offre veramente la possibilità di passare qualche ora di divertimento con il vostro TI 99.

IL GIOCO

Il gioco è molto semplice: la nostra astronave madre, di deposito sulla luna affinché con il vostro veicolo TERRA-ARIA possiate trasportare i minatori lunari, che lavorano per estrarre il minerale di Galinoide della superficie del satellite, da una cava all'altra.
Ovviamente questi minatori lavorano in luoghi impervi e scomodi da raggiungere.
Poiché il tempo è denaro, la ditta per cui lavorate vi dà un tempo limitato per il recupero, scaduto il quale, la vostra astronave, nel caso il recupero non fosse terminato, esploderà.
Ovviamente voi potete solo accelerare, a frenarvi penserà l'attrazione di gravità della luna, mentre l'atmosfera vi metterà in crisi per quanto riguarda gli spostamenti laterali.
È da notare che sulla superficie del pianeta, questi parametri sono soggetti continuamente a cambiamento per cui ci vorrà tutta la vostra abilità di pilota per portare a termine tutti i 16 quadri.

I COMANDI

Per giocare si hanno a disposizione 3 Moduli lunari. Il quadro non cambia mai tranne il colore delle montagne. Anche l'atmosfera e la gravità cambiano rispetto al quadro precedente.
Nel quadro a destra sono visualizzate le informazioni relative la gravità (GRAV), l'atmosfera (ATM), e il tempo (TM) e i moduli che restano ancora.
I minatori da recuperare sono quelle esili figure bianche che appaiono nel momento in cui il vostro modulo viene lasciato sulla superficie della luna. Per muovere il modulo bisogna usare i tasti con le frecce tranne quello con la «X» che non serve. L'abilità sta nel guidare il modulo gas in modo da stare in aria senza esagerare e soprattutto senza toccare nulla, pena la perdita del modulo.
Nel caso usciate dal quadro verso l'alto, è da considerare irrimediabile la perdita del modulo. Quando sarete riusciti ad avvicinarvi ad un minatore sarà sufficiente la pressione del tasto «Q» per fare salire a bordo il minatore.
Saliti tutti i minatori, si passa ad un altro quadro.
Un piccolo consiglio prima dell'inizio: per non perdere tempo conviene salvare il minatore sulla cima all'estrema sinistra del modulo; dopo aver effettuato il recupero uscite sulla sinistra e ricomparite sulla destra, quindi recuperate il minatore in profondità e infine tutti gli altri.
Buona fortuna!

MAURO MATTIAZZI

ROCKFELLER

In questo gioco voi impersonate un simpatico corvo che è stato imprigionato su una piramide di cubi colorati: scopo del gioco è quello di far saltellare il nostro eroe sulle facce superiori dei cubi fino a tingerle tutte di un solo colore.
Comunque Rockfeller non è il solo personaggio che saltella sui cubi: vi è Sam che ridiscende i cubi della piramide facendoli tornare al loro colore originario, per impedirglielo Rockfeller deve ingoiare Sam saltando sul suo cubo. Attenzione però, se è Sam a saltare sul vostro cubo sarete voi ad essere eliminati. Anche la palla verde, quando compare sullo schermo, va afferrata. Questo consente di bloccare il movimento di tutti gli altri personaggi presenti sulla piramide per un certo periodo di tempo. La palla rossa e quella nera vanno invece evitate, inoltre quella nera una volta giunta in fondo alla piramide fa uscire Coily, un serpentello avvolto su sé stesso che è l'unico in grado di risalire la piramide per dare la caccia a ROCKFELLER.

Quest'ultimo però ha un sistema efficacissimo per eliminare a sua volta Coily. Ai lati della piramide infatti si trovano due «dischi volanti»: sistemandosi su uno dei cubi laterali della penultima fila, e spiccando un salto verso l'esterno, Rockfeller può saltare su uno di tali dischi e venire così trasportato fino in cima alla piramide, inoltre ciò consente di eliminare tutti i nemici presenti sullo schermo, Coily compreso. Dopo gli schermi iniziali alle due piattaforme iniziali se ne aggiungono altre due, inoltre ogni piattaforma può essere utilizzata una volta sola per ogni fase di gioco:
Il movimento di Rockfeller avviene sempre in diagonale per cui il joystick viene utiliz-

zato in un modo diverso dal solito. Prendete il vostro joystick e mettetelo nella solita posizione, ora ruotatelo in senso orario di 45 gradi fino a formare un rombo con i suoi spigoli, in questo modo per far saltare Rockfeller in alto a sinistra basterà spingere il joystick in alto a sinistra e così per le altre direzioni. Se non avete i joystick o non volete usarli potete usare la tastiera utilizzando i seguenti tasti:

S — in alto a sinistra
D — in alto a destra
Z — in bassa a sinistra
X — in basso a destra

È molto importante assimilare al meglio i comandi per non commettere errori mortali: se Rockfeller salta per sbaglio dove non ci sono cubi, precipita nel vuoto e perde una vita.

SCHERMI

In ogni schermo occorre tramutare le facce superiori dei cubi nel colore finale; una volta completata una piramide, apparirà un nuovo schermo con una nuova serie di cubi.
Nei primi 3 schermi Rockfeller ha bisogno di un solo balzo su ogni cubo per passare dal colore iniziale (A) a quello finale (C). Negli schermi dal 4 al 6, il primo balzo di Rockfeller cambia il colore del cubo da uno stato iniziale A a uno intermedio B; per arrivare al colore finale C sarà poi sufficiente un secondo passaggio. Negli schermi dal 7 al 9, il colore passa direttamente da A a C dopo il primo passaggio per poi tornare ad A con il secondo. Per completare gli schermi in questa fase occorre quindi passare sui cubi un numero dispari di volte.

Negli schermi 10-12, il colore cambia da A a B a C, quindi nuovamente B, C, B, C e così via. Negli schermi 13-15, che si susseguono poi all'infinito, la faccenda si fa ancora più complicata: si raggiunge il colore finale C dopo due passaggi attraverso il colore iniziale A e quello intermedio B; basta però effettuare un balzo su un cubo con il colore C per tornare ad A per essere quindi costretti a ripetere i due passaggi precedenti.

Si ottengono 25 punti per ogni cubo che porta al suo colore conclusivo. Afferrare la palla verde dà diritto a 100 punti, mentre la cattura di Sam ne vale 300. Eliminare il serpente Coily dallo schermo con un disco fa guadagnare 500 punti, mentre il completamento di un intero schermo vale 3100 punti. I Rockfeller in dotazione sono 4: si perde una vita quando Rockfeller si scontra con uno dei suoi avversari oppure quando precipita nel vuoto. Si guadagna una vita-bonus ogni 5 schermi a partire dal 6. Alla fine del gioco per ricominciare bisogna premere RED o BACK.
Un'ultima nota riguarda il fatto che è possibile sospendere momentaneamente il gioco semplicemente premendo il tasto P fino a quando compare la scritta PAUSA; per riprendere il gioco basta premere poi un tasto.

GIAMPAOLO BOTTIN

ABBONATEVI
A TI 99
NEWSOFT.
CON SOLE
L. 95.000
RICEVERETE
11 NUMERI
DELLA RIVISTA
E IN PIU'
AVRETE
DIRITTO AGLI
SCONTI SUL
SOFTWARE
E
HARDWARE
OFFERTO DA
TI 99 MARKET

Per abbonarsi
basta inviare
un assegno
non trasferibile
o un vaglia
postale a:

NEWSOFT SNC
Via S. Jacini, 4
20121 MILANO

ITALIANO

CONIUGAZIONE DEI VERBI REGOLARI IN FORMA ATTIVA CON GLI AUSILIARI SIA ESSERE CHE AVERE

Il programma vuole simulare il metodo di apprendimento e memorizzazione degli elementi necessari alla conoscenza ed alla padronanza dei verbi regolari nella forma attiva.
Esso si articola nella titolazione dalla linea n. 100 alla 150, quindi si passa alla introduzione ed alla lettura del verbo che si vuole coniugare, assegnato alla variabile H$. Il verbo viene analizzato e scisso nelle due componenti: Radice e Desinenza, assegnate rispettivamente, a loro volta, alle variabili di stringa X$ e Q$.
A seconda della coniugazione selezionata si presenteranno dei menù differenti con colorazioni dello schermo diverse.
Tra le righe 260 e 270 si potrà inserire un goto 160, onde evitare che il computer accetti parole diverse da quelle terminanti in Are, Ere o Ire, maiuscole o minuscole che siano.
All'inizio di ciascuna sezione del programma, alle righe 270, 1630, 2660 alle variabili K$ e J$ vengono assegnate le desinenze singolare e plurare del participio passato. Segue la routine che visualizza lettera per lettera la scritta riguardante la coniugazione selezionata.

Per la prima coniugazione la scelta del tempo e del modo da coniugare avviene dietro la richiesta di scriverne esattamente la dicitura aggiungendo I per i tempi composti qualora il verbo richieda l'ausiliare ESSERE in quanto detto ausiliare concorre alla coniugazione della quasi totalità dei verbi intransitivi.
Per le altre due coniugazioni la scelta si opera su presentazione di un menù elencante tutti i tempi e modi e l'opzione per l'ausiliare Essere in seconda colonna. Il programma occupa tutta la memoria, per cui è impossibile aggiungere linee di istruzioni senza levarne altre.
Se si aggiungono le linee
151 GOTO 160
152 CALL CLEAR
153 PRINT "Non ho capito":::
265 GOTO 152
si dovranno cancellare le linee di rem, nella speranza che l'esecuzione del programma non venga interrotta con il messaggio MEMORY FULL IN...
Questo procedimento renderà più elegante il controllo dell'INPUT iniziale.
Il fine didattico di diversificare il modo di accedere ai tempi e modi delle tre coniugazioni, è quello di allenare all'uso della tastiera ed alla corretta ortografia oltre che alla corretta distinzione tra tempo e modo e tra verbi che richiedono l'ausiliare ESSERE invece che l'ausiliare AVERE.

Chi si meravigliasse nel vedere 6 righe di istruzioni nella linea dell'ON T GOTO, sappia che è possibile ottenere ciò scrivendo la linea di quattro righe nel modo più compatto possibile, battendo sull'ultimo spazio un carattere qualsiasi, premendo ENTER e quindi editando la linea appena scritta premendo SHIFT e Δ. Il cursore scivolerà fuori dall'ultima riga e quindi si potrà ancora scrivere sulla stessa linea.

Il programma tende a dimostrare come per saper coniugare con disinvoltura qualsiasi verbo, si debbano compiere due operazioni fondamentali: 1) saper riconoscere la coniugazione di appartenenza del verbo dalla desinenza, come fa il programma, 2) mandare a memoria tutte le desinenze.

Va da sè che volendo si possono visualizzare solo le desinenze, inserendo la sola desinenza ARE, ERE o IRE.

FERDINANDO GIGLIOTTI

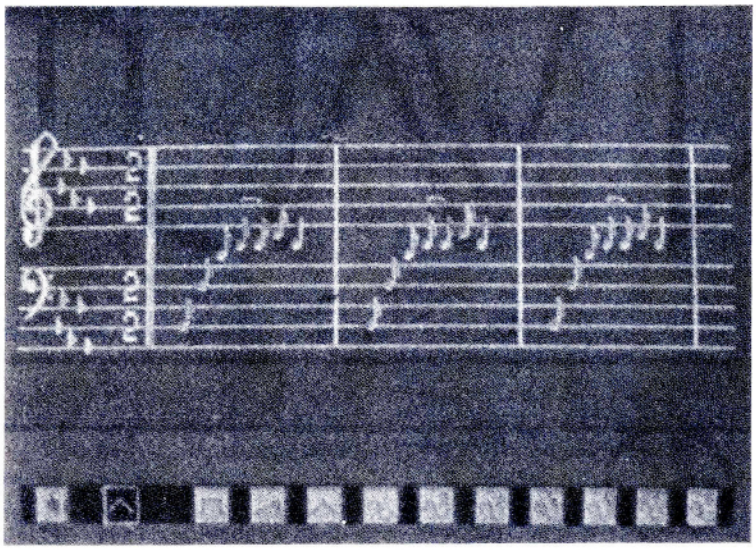
CALORIE

Questo programma vi permette di sapere abbastanza rapidamente se i vostri pasti contengono troppe calorie, oppure se contengono in misura sufficiente tutte quelle sostanze che sono indispensabili per il buon funzionamento del nostro corpo, cioè quello che si chiama il bilanciamento della dieta.
Per prima cosa il computer vi domanderà che genere di attività svolgete; questo per stabilire il vostro fabbisogno giornaliero di calorie; ovviamente chi svolge un lavoro che richiede una grande attività fisica avrà un fabbisogno di calorie molto superiore di chi invece fa un lavoro sedentario.
Poi vi verrà chiesto se intendete prendere in esame solo una pasto oppure tutto il cibo ingerito durante il giorno.
Seguirà un elenco di alimenti, ad ogni domanda del computer occorre introdurre la quantità ingerita espressa in grammi ed è molto importante cercare di essere più precisi possibili sulla quantità.
Se un alimento non compare nel pasto è sufficiente premere ENTER per passare al seguente.
Terminato l'elenco degli alimenti comparirà un quadro riassuntivo con tutte le qualità totali delle varie sostanze chimiche presenti negli alimenti che sono stati introdotti nel programma.
Premendo ancora ENTER si avranno le conclusioni sul bilanciamento, il computer cioè vi dirà quali sostanze risulteranno in eccesso e quali invece risulteranno in quantità inferiore alla minima giornaliera consentita.
Nel caso che alcuni minerali o vitamine risultino presenti in quantità troppo bassa, consultando la seguente tabella voi potrete sapere quali cibi dovrete introdurre nei vostri pasti per raggiungere un bilanciamento più adeguato.
I dati relativi alla composizione chimica degli alimenti sono stati tratti dalle tabelle dell'Istituto Nazionale per la Nutrizione.

MAURIZIO CAPPA

Elemento	Si trova in:
PROTIDI	carni, pesci, uova, latte, formaggi
GLICIDI	cereali, legumi, frutta, ortaggi
LIPIDI	burro, panna, lardo, olii
CALCIO	latte e formaggi
FERRO	frattaglie, turlo d'uovo, carne
POTASSIO	frutta, ortaggi, legumi, carni
VIT. B1	noci, castagne, legumi, fegato, rene, cuore, carne di maiale
VIT. B2	fegato, rene, cuore, latte, uova
VIT. PP	lievito, carne, fegato
VIT. A	fegato, rene, burro, formaggi, carote, pomodori, albicocche, pesche, mandarini

MUSICAL AIDER



Il programma in questione si può definire multiobbiettivo in quanto diverse possono essere le finalità di impiego: diletto, utility, ecc.
Esso è principalmente diretto a tutti quei lettori, come il sottoscritto, che suonano preferibilmente ad orecchio e, pur disponendo di rudimenti di teoria musicale, faticherebbero non poco allorchè si rendesse necessario individuare la esatta sequenza di note che costituisce una melodia.
Tale operazione peraltro è fondamentale in tutti quei casi nei quali si desideri inserire colonne sonore in un programma autoprodotto che, va detto per inciso, acquista sempre un fascino particolare se è accompagnato da un adeguato commento musicale.
Nello stesso tempo va altresì offerta la possibilità di riascoltare la melodia digitata con vari timbri sonori, con diversa velocità, intervenendo con tutte le correzioni e modifiche del caso senza essere costretti a ripartire ogni volta dall'inizio.

IL FUNZIONAMENTO

Dopo il RUN appare il titolo accompagnato da una introduzione musicale (brano di una nota fuga) seguito da una breve descrizione sull'uso del programma.
La tastiera viene utilizzata alla stessa stregua della ottava di un pianoforte con i tasti egualmente disposti; si utilizzano infatti i tasti DO/SI corrispondenti alle lettere Z/M in basso a sinistra. I corrispondenti diesis sono collocati sulla linea di tastiera immediatamente superiore, come per i tasti neri di un pianoforte.
Generata la melodia (è possibile in qualunque momento modificare la ottava premendo i tasti 1/3 in quanto si dispone sino a tre ottave) si può scegliere fra diverse operazioni:
— Ripetizione del brano (tutto o in parte)

ANALISI LISTATO

N. Riga	Descrizione
130 - 310	Assegnazioni iniziali
320 - 460	Presentazione grafica
470 - 570	Presentazione sonora
580 - 690	Istruzioni per l'uso del programma
700 - 840	Introduzione ottava desiderata
850 - 1330	Attivazione tastiera
1340 - 2160	ROUTINE comprendente la selezione funzione e l'ascolto
2230 - 2400	ROUTINE di stampa (o pagina video)
2410 - 2510	ROUTINE CAMBIO PAGINA
2540 - 2860	ROUTINE transcodifica tasti/frequenze.

VARIABILI PRINCIPALI

GR$(13)	Identificazione grafica nota
OT$(3)	Identificazione grafica ottava
M(250)	Frequenza nota memorizzata
D(250)	Durata nota memorizzata
O	Ottava prescelta
U	Numero battute
V	Timbro prescelto in ascolto
NOT(13)	Codifica tasto digitato
Z(50)	Codifica interna

— Modifica del brano
— Stampa della sequenza del tipo di nota, della frequenza, della durata.

In fase di digitazione della melodia, va aggiunto che è possibile in qualunque momento correggere la ultima o le ultime note introdotte premendo il tasto 9 e ripetendo la battuta; la durata di pressione dei tasti è memorizzata.

La audizione è possibile su 5 diverse velocità e con diversi timbri:
— monotonale
— fisarmonica
— organo classico
— organo moderno

È così possibile riascoltare, anche se si è suonatori impacciati che fanno molti errori, la melodia introdotta, con perfetto tempo e sequenza.

La stampa della sequenza delle note, con relativa frequenza e durata, è particolarmente utile per introdurre, in una istruzione di DATA il commento musicale a programmi da noi stessi prodotti.

Per quanto riguarda le caratteristiche del programma, si rimanda alla allegata tabella, sottolineando solo che il largo impiego di vettori (necessario per velocizzare i vari controlli), lascia disponibile un Kb. circa. Una apparente complicazione del programma, con le sue transcodifiche interne, è una conseguenza di ciò.

R. DRUSIANI

Avete sviluppato un programma originale, divertente, utile? Volete vederlo pubblicato su uno dei prossimi numeri di

TI-99 NEWSOFT?

Leggete allora con attenzione a pag. 31-32 e mandateci le vostre proposte

TI 99 NEWSOFT

MARZO 1985 L. 10.000

BASIC
Master Control Program

EXTENDED + 32 K
VDP senza segreti

TI 99/4A - Le RAM

IN SOFTWARE & SOFTWARE:
Hot Dog - Strike Defence - Writer
Loader - Family Spreadsheet - Nibbler

SOMMARIO

□ **TI BASIC**
Le ultime notizie sul linguaggio TI BASIC — pag. 2

□ **EXTENDED**
Approfondiamo l'argomento della protezione del software — pag. 3

□ **TI 99/4A**
Vi spieghiamo cos'è e come funziona una RAM — pag. 5

□ **EXTENDED** + ESPANSIONE 32 K
Come si può tradurre un programma ASSEMBLY in L/M — pag. 7

□ **WRITER LOADER**
Un programma che carica in memoria sotto-programmi in L/M — pag. 9

□ **STATISTICA**
Il teorema del limite centrale — pag. 11

□ **M.C.P.**
Il Master Control Program - Un sistema di archiviazione per programmi — pag. 12

□ **HOT DOG**
Uno dei giochi più divertenti ed originali ora anche per il TI 99 — pag. 15

□ **STRIKE DEFENCE**
Non lasciate che il nemico distrugga il vostro deposito di carburante nucleare. Difendetevi!! — pag. 16

□ **FAMILY SPREAD SHEET**
Un tabellone elettronico per la contabilità familiare, per ordinare le entrate e le uscite — pag. 18

□ **MISSILE COMMAND**
La traduzione per il TI 99 dell'omonimo famoso gioco da arcade — pag. 20

□ **NIBBLER**
Proponiamo questo «bets-seller » in versione TI BASIC — pag. 22

□ **DOSSIER**
Il computer, questo conosciuto... — pag. 23

□ **RECENSIONI**
Parliamo del programma «Decathlon» — pag. 25

□ **LA VIGNETTA** — pag. 26

□ **LA POSTA** — pag. 27

□ **PICCOLI ANNUNCI** — pag. 28

□ **COLLABORATE CON NOI** — pag. 31

□ **TI 99 MARKET** — pag. 33

TI 99 NEWSOFT
ANNO 2 - Numero sette
MARZO 1985
MENSILE
Direttore responsabile:
ANTONIO C. LOSITO
Coordinatore Editoriale:
VINICIO CIVININI
Direttore Amministrativo:
ENRICO CALLERIO
Art Director:
BEPPE RE-FRASCHINI
Illustrazioni:
MARIA TERESA COCOZZA
Segreteria:
EMANUELA CHIAPELLO
CRISTINA FRISON
Redazione:
FIORENZA AURIEMMA
DARIO BAREZZI
Collaboratori:
SERGIO BORSANI,
GIANPAOLO BOTTIN,
DANIELE CATALFAMO,
PAOLO CIVARDI,
STEFANO DELLI PONTI,
FRANCESCO MACALI,
MAURO MATTIAZZI,
EZIO MONTINI,
LEVIO PEROTTI,
RICCARDO ROSSI,
MARCO SQUINTANI,
ALBERTO STRAFILE,
SEBASTIANO TOMASELLO
TI 99 NEWSOFT è una pubblicazione della **NEWSOFT snc**, via Stefano Jacini 4 - 20121 MI
Tel. 02/807464 - Telex 324284
REGISTRAZIONE AL TRIBUNALE DI MILANO N. 323 DEL 7.7.84

COPYRIGHT NEWSOFT snc
TUTTI I DIRITTI RISERVATI
Manoscritti e foto originali, anche se non pubblicati, non si restituiscono ed è vietata la riproduzione anche parziale di testi, fotografie e listati.

Pubblicità: AV STUDIO
Via Giuseppe Meda, 23
Tel. 02/8357155
Abbonamenti: l'abbonamento annuo a 11 numeri di *TI 99 NEWSOFT* costa L. 95.000 per l'Italia; L. 160.000 per l'Europa (spedizione aerea); L. 200.000 per tutti gli altri paesi.
L'importo, per assegno non trasferibile o per vaglia postale, va inviato a NEWSOFT snc, via S. Jacini 4 - 20121 Milano.
Stampa: CGB, Albegno di Treviolo (BG)
Concessionaria per la distribuzione:
EUROSTAMPA Srl - C.so Vittorio Emanuele 111 - 10123 Torino - Tel. 011/538166-7

L'editore non si assume responsabilità alcuna sul contenuto degli articoli e/o software inviati.
Per gli articoli firmati e/o siglati da collaboratori esterni, la redazione si assume la responsabilità prevista dalle leggi sulla stampa.

Tutti possono collaborare a TI 99 NEWSOFT, sia con articoli che con programmi. Scriveteci o telefonateci in redazione per avere tutti i ragguagli.

ULTIME NOTIZIE

SUL LINGUAGGIO TI BASIC

Su questo linguaggio ne sono state dette di cotte e di crude: dalle euforiche esaltazioni per l'eccezionale grafica alle secche e accidiose critiche alla sua lentezza.
Non intendiamo ora aprire un dibattito e neppure sentenziare una verità assoluta, ma solo darvi qualche notizia o consiglio per sfruttare le capacità latenti di questo nostro tanto discusso linguaggio.

CALL KEY

Questo affascinante comando non finirà mai di stupire quanti ne fanno uso. Provate infatti a digitare questo semplice programmino:

```
100 CALL KEY (3, K, S)
100 IF S= OTHEN 100
120 PRINT K
130 GOTO 100
```

Fatto? Ora date il RUN e provate a premere qualche tasto. Sul video appariranno degli strani numeri che altro non sono se non i valori ASCII del tasto che avete premuto. Provate quindi a battere lo stesso tasto, prima come maiuscola e poi come minuscola e state a vedere. Il Texas non farà alcuna differenza tra lettere maiuscole e minuscole, questo perché l'unità di tastiera 3 era esclusiva del vecchio TI 99/4 che come certo saprete non aveva le lettere minuscole (o compresse che siano). Spiegato ora l'arcano? Andate adesso a sostituire la linea 100 con una CALL KEY (4, K, S) e la linea 120 con INPUT K$ e premete un tasto dopo aver fatto ripartire il programma. Apparirà un punto di domanda (causato dalla INPUT). Scrivete quello che volete e cercate in seguito di fare qualche correzione. Il cursore impazzito si rifiuterà di tornare indietro, di inserire, di cancellare, addirittura di bloccare il programma con il comando di BREAK (FCTN 4).

Niente allarmismi; la tanto temuta ribellione dei computer non è ancora in atto, semplicemente la tastiera da BASIC è diventata PASCAL! Ne volete conferma? Sostituite i vari comandi con la seguente tabella e tutto tornerà normale (si fa per dire).

```
ENTER  = CTRL M
FCTN S = CTRL H
FCTN D = CTRL I
FCTN E = CTRL K
FCTN X = CTRL J
FCTN 1 = CTRL C (DEL)
FCTR 2 = CTRL D (INS)
FCTR 3 = CTRL G (ERASE)
FCTN 4 = CTRL B (CLEAR)
```

Per bloccare i due programmi usate dunque il solito modo per il primo, CTRL B per il secondo. Nella speranza di avervi reso un gradito servizio, ci congediamo dandovi appuntamento al prossimo numero.

MARCO SQUINTANI

PROTEGGERE
IL SOFTWARE

La scorsa volta abbiamo trattato a grandi linee l'argomento della protezione del software, ora lo approfondiremo. Vedremo in quale modo sia possibile mettere in opera, ciò che abbiamo detto precedentemente.
Il metodo più sicuro di protezione, tra quelli già trattati, rimane sempre quello di salvare il programma con la formula SAVE CS1, PROTECTED, ma ciò può risultare scomodo, in quanto non permette alcun tipo di intervento sul listato, quindi il proprietario del programma è sempre costretto a tenere due copie del suo programma.
Come promesso, vi proponiamo due piccoli sottoprogrammi che vi chiariranno le idee, per quanto riguarda gli altri metodi di protezione che vi abbiamo proposto. Il sottoprogramma contrassegnato con il numero uno, funziona sia in ambiente EXT.BASIC, che in EXT.BASIC + 32K.

Esegue un controllo con una CALL GCHAR alla riga 24 ed alla colonna 2.
Generalmente in questa posizione si trova il segno di pronto del computer, costituito dal carattere «>», corrispondente al codice ASCII (31).
Se il nostro piccolo sottoprogramma verifica che questo carattere è presente sullo schermo alle coordinate indicate, blocca il sistema, con una conseguente perdita del programma.
In questo modo, possiamo obbligare l'utilizzatore del programma a mandarlo in «RUN» solamente con una RUN CS1.
Per impedire il successivo listato o interruzione, possiamo includere nel programma le istruzioni ON BREAK NEXT e ON WORNING NEXT.
ON BREAK NEXT, può impedire il «breccaggio» del programma, la seconda impedisce che il programma si fermi, nel caso siano stati inseriti erratamente dei dati nelle INPUT. Per ulteriori chiarimenti su queste istruzioni vedere manuale EXTENDED capitolo 4, alla pagine 130 e 137.
Per comodità abbiamo usato una CALL, per chiamare la subroutine di controllo. Il nome della CALL può essere facilmente cambiato con qualcosa di più «innocente», come START, PRESENT ecc., a vostra discrezione.
Questo metodo di protezione non è certo invulnerabile, anzi se dobbiamo dire la verità, esso presenta parecchi buchi, se così li possiamo definire, ma può essere utile come una ulteriore protezione.
Vi consigliamo di mettere molte CALL PEEK nel programma, per non insospettire l'utilizzatore del programma.
Potete ad esempio utilizzare il generatore di numeri ca-

suali, descritto nel numero di settembre nella rubrica dedicata all'EXTENDED BASIC.
Vi domanderete, perché noi ci diamo tanto da fare per trovare nuovi metodi di protezione, diversi da quello fornito dalla casa costruttrice.
La ragione è che in linea generale NESSUN metodo si può definire SICURO.
Infatti i programmi salvati con l'opzione PROTECTED sono sproteggibili. Per ovvi motivi non possiamo rivelarvi questo «segreto».
Se scoprite qualche nuovo metodo di protezione, fatecelo sapere, spedendo i vostri lavori, accuratamente commentati, alla nostra redazione.
Per il sottoprogramma numero due, valgono le regole precedentemente dette, però esso funziona solamente con la configurazione EXT.+32K.
Per motivi di stampa la riga 200, presenta un piccolo errore, i «::» due punti, non devono essere spaziati, la riga 200 dovrà essere scritta nel seguente modo:

```
200 IF A<>32 THEN
      PRINT ::::::
```

EZIO MONTINI

```
1  !+-------------------------------+
2  !!     SOTTOPROGRAMMA          !
3  !!     PROTEZIONE DEL          !
4  !!     SOFTWARE -N.1-          !
5  !+-------------------------------+
100 CALL CONTROLLO
110 .
120 .
130 .
140 !     LINEE DI PROGRAMMA
150 .
160 .
170 .
180 SUB CONTROLLO
190 CALL GCHAR(24,2,A)
200 IF A<>32 THEN CALL PEEK(-26624,A)
210 SUBEND
```

```
1  !+-------------------------------+
2  !!     SOTTOPROGRAMMA          !
3  !!     PROTEZIONE DEL          !
4  !!     SOFTWARE -N.2-          !
5  !+-------------------------------+
100 CALL CONTROLLO
110 .
120 .
130 .
140 !     LINEE DI PROGRAMMA
150 .
160 .
170 .
180 SUB CONTROLLO
190 CALL GCHAR(24,2,A)
200 IF A<>32 THEN PRINT :: :: ::
210 SUBEND
```

LE RAM

Proseguiamo il cammino nell'affascinante mondo dei componenti del nostro computer, soffermandoci questa volta a considerare le memorie di lavoro. Fino ad ora abbiamo capito la grandissima importanza che riveste il microprocessore all'interno delle operazioni svolte dall'hardware e abbiamo altresì visto che esso necessita di tutta una serie di elementi «collaboratori» che gli forniscano i dati, li trasportino, li memorizzino, ecc. ecc.
A proposito di memorizzazione, dando un'occhiata alla breve storia del computer, si è visto come le prime memorie consistessero in costosi e ingombranti blocchi di ferrite. Tali blocchi avevano il pregio di non essere volatili, cioè di poter conservare i loro dati anche a spegnimento totale della macchina.
Oggigiorno le memorie — circuiti integrati ad alto grado di miniaturizzazione — non hanno più questo pregio, ma possiedono, secondo le funzioni che sono chiamate a svolgere, differenti caratteristiche. Tratteremo per il momento le sole RAM (poi vedremo che cosa sono), riservandoci di spiegare successivamente l'uso delle ROM, PROM, e delle EPROM.

LE RAM

La sigla RAM, d'obbligo nel campo dell'elettronica, sta a significare Random Access Memory, che risulta più comprensibilmente alle nostre orecchie come «memoria ad accesso casuale».
All'accensione del computer, le RAM si presentano come dei cassetti vuoti che è possibile riempire con dati o programmi di nostra scelta, che possono essere introdotti di volta in volta da tastiera o essere acquisiti da dispositivi di memoria di massa (per esempio, il nostro fedele CSI). Con molta memoria RAM a disposizione sarà possibile digitare ed eseguire sul calcolatore programmi sempre più lunghi e versatili o gestire quantità maggiori di dati. Ogni carattere, come già visto, occupa un byte, perciò dire che il nostro Texas ha una capacità di memoria in configurazione base di 16K significa che può contenere fino a 16.000 caratteri. Il che non è male.
La memoria di base è, per quei pochi che non lo sapessero, espandibile con 32K aggiuntivi (e forse più sperando nel futuro).
Da notare, sotto l'aspetto puramente economico, la grande utilità di queste espansioni che permettono di limitare l'investimento iniziale e acquistare memoria aggiuntiva solo quando se ne abbia veramente bisogno.
Abbiamo già detto che le RAM, in quanto non consentono di conservare i dati una volta spento il computer (a parte la piccola eccezione fatta per la mini-memory), necessitano di una certa procedura obbligata. Bisogna infatti avere l'accortezza, ogni qual volta si digiti un programma che si intende riutilizzare, di registrare quanto fatto su un dispositivo di memoria di massa con gli arcinoti comandi SAVE CS1 o SAVE DSK1. Non è a questo punto superfluo controllare se il programma o i dati sono stati esattamente salvati rispondendo Y (sì) alla domanda CHECK TAPE? (controllo il nastro?) del computer.
Da un punto di vista strettamente tecnico, le RAM si dividono in statiche e dinamiche.

L'unica differenza che le distingue è che, mentre le prime mantengono i loro dati fino a spegnimento, le seconde hanno bisogno di un circuito che «rinfreschi» loro la memoria ogni pochi millisecondi.
Che cosa differenzia tra loro le RAM? Il cosiddetto tempo di accesso, o meglio il tempo necessario alla memoria per reperire un dato al suo interno. Se la velocità di accesso delle RAM è supersonica, ma quella del microprocessore no, inutile dire che il risultato finale sarà comunque influenzato negativamente.
Per «ripulire» le RAM senza dover spegnere tutto quanto, in molti computer sono attivati dei tasti di RESET (riposizionamento). Nel TI 99 sono presenti anche da basic due comandi che producono lo stesso disarmante effetto dell'FCTN + (Quit); uno, il NEW, spazza le RAM senza uscire dal linguaggio; l'altro, il BYE, porta addirittura al titolo principale. È sempre meglio usare questi due comandi, tranne nel caso in cui si debba uscire da un modulo di SOLID STATE SOFTWARE.
Per arricchire di questi pochi dati le vostre personali memorie permanenti, provate ora a cimentarvi al computer con il programma TEACH BIT, senza possibilmente farvi prendere in castagna.

MARCO SQUINTANI

DICHIARAZIONE DEI REDDITI 1984

E' disponibile un programma
che finalmente vi guiderà alla determinazione
del reddito per i Mod. 101 e 740.

Il programma gira in Extended Basic
e costa Lit. 35.000 + spese di spedizione.

PER RICHIEDERLO
UTILIZZATE LA CARTOLINA
TI 99 MARKET

IMPARIAMO IL LINGUAGGIO MACCHINA

Questa volta vedremo come praticamente si possa tradurre un programma ASSEMBLY in uno in Linguaggio Macchina, spiegando come, passo per passo, si può tradurre. Il programma di figura 1 è la sorgente del programma WRITERLOADER già tradotto in LM e presente sulla cassetta.
Avrete notato che il listato è diviso in due parti distinte, una strettamente numerica e una solo mnemonica (cioè fatta con simboli).
La parte simbolica rappresenta il programma scritto in ASSEMBLY, che è la prima stesura del programma.
In genere si utilizza questo linguaggio perché è più facile da ricordare.
La parte numerica esadecimale è a sua volta suddivisa in tre sezioni. Partendo da sinistra troviamo:

1) numeri di linea
2) locazione di memoria
3) traduzione delle istruzioni

La parte 1 non è di nessuna importanza ai fini del programma ed è trascurabile.
La parte 2 rappresenta le locazioni di memoria occupate dall'istruzione assembly, che possono essere cambiate, osservando alcune regole.
Poniamo il caso di partire dalla locazione dec. 9460, alla locazione 9460 inseriremo il valore, tradotto in decimale, del primo BYTE, non della prima word.
Infatti la word in questione contiene il valore esadecimale >FFF che diviso in BYTE è >FF >FF. Tradotto è uguale a 256,256.
Questa operazione va eseguita per ogni word del nostro programma, e incrementando sempre la locazione di memoria, in questo caso avremo:

locazione decimale	*valore inserito*
9460	256
9461	256
ecc.	ecc.

La parte 3 é la più importante, perché contiene la traduzione dei codici mnemonici, che andranno tradotti e inseriti nella maniera sopra indicata.
Nella prima colonna della parte simbolica, troviamo i label (etichette di indentificazione) che servono per poter richiamare i sottoprogrammi assembly, ovviamente con il nome indicato.
Ad esempio, la routine WRITER inizia alla locazione esadecimale >0002 che tradotta in decimale è 0,2.
Attenzione, come spiegato la volta scorsa il nome della routine va inserito alla locazione dec. 16376 e seguenti e la locazione deve essere tradotta in nibble e inserita.
Se il programma parte dalla locazione dec. 9460, la label si troverà alla loc. 9462, che a sua volta tradotta in nibble è 36,246.
Le modalità per l'inserimento sono state spiegate accuratamente la scorsa volta, quindi non ci soffermeremo su

questo.
La seconda label si trova a 9460+20=9480, che tradotta è 37,8.
Essa sarà inserita a partire dalla locazione dec. 16376-8=16368, perché 8 byte sono stati occupati dalla precedente.
La label LOOK non va inserita nelle DEF come le precedenti, perché non rappresenta il punto di entrata di nessun programma.
Essa si può definire come una variabile che contiene il valore della locazione di memoria dove è allocato il mini buffer in cui è depositato, nel nostro caso, il valore di ritorno della routine LOADER.
Analizziamo la principale istruzione del programma.
Load Immediate mnemonico LI muove il contenuto della word in un registro.

traduzione esadecimale	*mne.*	*reg.*	*valore word*
>02 00 >00 00	LI	R0,	0000
>02 01 >00 FF	LI	R1,	00FF

Essa ha come traduzione >02 ed è il primo byte, poi troviamo il numero del registro interessato dall'operazione, nella seguente word il valore passato. Andiamo a vedere cosa è successo in memoria una volta eseguita questa operazione:

```
  word del registro prima della LI
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  primi 8 bit  | secondi 8 bit |
+---------------+---------------+
|   prima word  |  seconda word |
+---------------+---------------+

  word del registro dopo la LI
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0|0|0|0|0|0|0|0|1|1|1|1|1|1|1|1|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  primi 8 bit  | secondi 8 bit |
+---------------+---------------+
|   prima word  |  seconda word |
+---------------+---------------+
          esa.> 00FF
```

Le altre istruzioni mnemoniche le tratteremo in seguito. Se avete già delle esperienze, o volete che il contenuto della rubrica cambi, scrivetecelo, in questo modo potremo soddisfare meglio le vostre esigenze.

EZIO MONTINI

GIOCHIAMO CON IL TI 99/4A

Un libro che non deve assolutamente mancare nella vostra libreria.

Potrete riceverlo a casa vostra a sole **L. 9500** + Spese di spedizione.

Per ogni richiesta di contrassegno, utilizzate il tagliando qui sotto spedendolo a:

NEWSOFT snc
via S. Jacini, 4
20121 MILANO

Vogliate spedirmi in contrassegno di L. 9500 + spese il libro: «GIOCHIAMO CON IL TI 99/4A»

Cognome .....

Nome .....

Indirizzo .....

.....

Cap. ..... Loc. .....

«WRITER LOADER»
VDP SENZA SEGRETI

Il programma di nome WRITER LOADER che trovate sulla cassetta, deve essere caricato con la formula OLD CS1, poi mandato in esecuzione con il classico RUN.
Questo programma carica in memoria un set di sottoprogrammi in linguaggio macchina, eseguibili con delle CALL LINK.
Esso è in grado di autogestire le CALL LINK, introducendo i valori esatti, nel programma in linguaggio macchina, per ottenere ciò che vogliamo.
Il programma è suddiviso in tre parti principali, che sono richiamate da delle CALL:

1) CALL WRITELOAD
2) CALL WRITER
3) CALL LOADER

La prima si occupa di caricare il programma in LM che è in grado di fornirci sottoprogrammi WRITER e LOADER. Questi sottoprogrammi ci permettono di leggere e scrivere liberamente in tutta la VDP del nostro computer.
Alla richiesta locazione, noi dovremo inserire la locazione della VDP ram dove vogliamo intervenire, successivamente ci sarà richiesto il valore che si desidera inserire.
Il programma è strutturato

```
0001              ************************************
0002              * IMPARIAMO A PROGRAMMARE IN LM *
0003              *                                  *
0004              *      OGGETTO E SORGENTE          *
0005              *              DEL                 *
0006              *                                  *
0007              * Programma : V D P WRITERLOADER*
0008              * Version   : EXT+32k              *
0009              * Autor     : EZIO MONTINI         *
0010              ************************************
0011
0012                              DEF WRITER,LOADER
0013
0014 0000 0000    LOOK    DATA >0000
0015
0016
0017 0002 C28B    WRITER  MOV  R11,R10
0018 0004 0200            LI   R0,>0000
     0006 0000
0019 0008 0201            LI   R1,>0000
     000A 0000
0020 000C 0420            BLWP @>2020
     000E 2020
0021 0010 C2CA            MOV  R10,R11
0022 0012 045B            B    *R11
0023
0024
0025 0014 C28B    LOADER  MOV  R11,R10
0026 0016 0200            LI   R0,>0000
     0018 0000
0027 001A 042D            BLWP @>2028
     001C 2028
0028 001E C801            MOV  R1,@LOOK
     0020 0000'
0029 0022 C2CA            MOV  R10,R11
0030 0024 045B            B    *R11
0031
0032                              END
0000 ERRORS
```

per inserire solamente un dato per ogni richiesta, ma è facilmente modificabile.

Esso può inoltre essere utilizzato in altri programmi da voi elaborati, estraendo solo la routine da voi desiderata.

Per poter utilizzare le CALL WRITER e CALL LOADER è necessario caricare prima la main ruotine CALL WRITELOAD.

Disponiamo di due sottoprogrammi in LM, richiamabili con:

a) CALL LINK («WRITER») per scrivere in VDP
b) CALL LINK («LOADER») per leggere la VDP

La seconda parte gestisce i dati per far funzionare correttamente la routine in LM WRITER, e richiede che la locazione di memoria sia tradotta in nibble; le righe 350, 360 dividono la locazione di memoria denominata LOC, in MSB in LOCA, e LSB in LOCB, introducendoli all'indirizzo decimale 9466 e 9467 prima di eseguire la CALL LINK.

Il valore da inserire in quella locazione viene scritto alla locazione decimale 9470, ed è contenuto nella variabile VALO.

Il valore da inserire non deve mai essere negativo e non deve mai superare 256.

La terza parte infine si occupa di leggere i valori contenuti alla locazione desiderata.

Anche in questo caso la locazione di memoria deve essere tradotta in nibble.

Il programma basic esegue automaticamente tale operazione, alle righe 460, 470. Detti valori MSB in LOCA e LSB in LOCB, sono introdotti nella locazione decimale 9484. E il sottoprogramma, una volta mandato in esecuzione, ci restituisce il valore di quella locazione nella locazione decimale 9460.

Esso andrà in seguito letto con una CALL PEEK e il risultato sarà messo nella variabile numerica RIS.

EZIO MONTINI

```
100 !+------------------------+
110 !! PROGRAMMA PER SCRIVERE!
120 !! LEGGERE LA  V D P RAM !
130 !! Autore  : EZIO MONTINI!
140 !! Versione: EXT + 32Kram!
150 !+------------------------+
160 CALL CLEAR
170 FOR A=0 TO 14 :: CALL COLOR(A,16,1):: NEXT A :: CALL SCREEN(5)
180 CALL CLEAR
190 DISPLAY AT(12,1):"VUOI CARICARE IL LM(S/N) S" :: ACCEPT AT(12,26)VALIDATE("S
N")SIZE(-1):A$
200 IF A$="S" THEN CALL WRITELOAD
210 CALL CLEAR
220 DISPLAY AT(3,1):"MENU  GENERALE  WRITERLOADER            BY
    MONTINI EZIO"
230 DISPLAY AT(9,1):"W -WRITE VDP LOCATION MEMORY"
240 DISPLAY AT(11,1):"L -LOAD  VDP LOCATION MEMORY"
250 DISPLAY AT(13,1):"E -END   VDP WRITERLOADER"
260 DISPLAY AT(22,7):"(W/L/E)" :: ACCEPT AT(22,15)VALIDATE("WLE")SIZE(-1):B$
270 IF B$="W" THEN CALL WRITER
280 IF B$="L" THEN CALL LOADER
290 IF B$="E" THEN CALL CLEAR :: END
300 GOTO 210
310 SUB WRITER
320 CALL CLEAR
330 DISPLAY AT(9,1):"LOCAZIONE DECIMALE " :: ACCEPT AT(9,20)SIZE(-5)VALIDATE(NUM
ERIC):LOC
340 DISPLAY AT(11,1):"VALORE DEC. (0-256)" :: ACCEPT AT(11,20)SIZE(-3)VALIDATE(D
IGIT):VALO
350 LOCA=INT(LOC/256)
360 LOCB=INT(((LOC/256)-LOCA)*256)
370 CALL LOAD(9466,LOCA,LOCB,"",9470,VALO)
380 CALL LINK("WRITER")
390 DISPLAY AT(23,4):"OK. ANCORA (S/N) S" :: ACCEPT AT(23,21)SIZE(-1)VALIDATE("S
N"):C$
400 IF C$="S" THEN 320
410 SUBEND
420 SUB LOADER
430 CALL CLEAR
440 DISPLAY AT(9,1):"LOCAZIONE DECIMALE " :: ACCEPT AT(9,20)SIZE(-6)VALIDATE(NUM
ERIC):LOC
450 DISPLAY AT(11,1):"VALORE LETTO ... = ";RIS
460 LOCA=INT(LOC/256)
470 LOCB=INT(((LOC/256)-LOCA)*256)
480 CALL LOAD(9484,LOCA,LOCB)
490 CALL LINK("LOADER")
500 CALL PEEK(9460,RIS)
510 DISPLAY AT(11,1):"VALORE LETTO ... = ";RIS
520 DISPLAY AT(23,4):"OK. ANCORA (S/N) S" :: ACCEPT AT(23,21)SIZE(-1)VALIDATE("S
N"):C$
530 IF C$="S" THEN 430
540 SUBEND
550 SUB WRITELOAD
560 CALL INIT
570 CALL LOAD(16368,76,79,65,68,69,82,37,8)
580 CALL LOAD(16376,87,82,73,84,69,82,36,246)
590 CALL LOAD(8194,37,26,63,240)
600 CALL LOAD(9460,0,0,194,139,2,0,0,10,2,1,178,0,4,32,32,32,194,202,4,91,194,13
9)
610 CALL LOAD(9482,2,0,0,0,4,32,32,40,200,1,36,244,194,202,4,91,40,183)
620 SUBEND
```

IL TEOREMA DEL LIMITE CENTRALE

Definizione:
Se da una popolazione apparentata avente media μ e deviazione standard σ vengono tratti, in modo casuale, campioni di grandezza n1, le medie di questi campioni costituiranno una popolazione di media μ e la cui deviazione standard tende a divenire uguale aσ / \ n1 quando n1 tende all'infinito.

Che la popolazione vera sia o non sia normale, le medie campionarie tenderanno sempre ad assumere una distribuzione normale quando n1 tende all'infinito.

Rileviamo che:

1) Se la popolazione è distribuita secondo la funzione di densità di probabilità gaussiana, la distribuzione delle medie campionarie è sempre gaussiana, qualunque sia la dimensione del campione.

2) La varianza di una media campionaria σ 2/Y è funzione della varianza della popolazione e della numerosità n1 del campione.

$$\sigma \frac{2}{y} = \frac{\sigma^2}{n1}$$

e la quantità

$$\frac{\sigma}{y} = \frac{\sigma}{\sqrt{n1}}$$

viene definita Errore Standard della media.

L'approsimazione del modello proposto dal teorema del limite centrale è abbastanza buona quando n1 è superiore, o uguale a 5 mentre è del tutto accettabile per n1 >= 15.

È importante ribadire che il teorema va solamente impiegato quando i dati originali non siano disponibili, nel caso questi invece esistano è d'obbligo usare il programma, già pubblicato su questa rivista, per il calcolo della deviazione standard, molto più affidabile in quanto fa uso di tutti i dati a disposizione a differenza di quello odierno che si limita ad usare alcune medie campionarie.

(Per i più volonterosi sarebbe interessante comparare, su una serie di dati, i valori ottenuti e decidere quindi l'azione più opportuna per ottenere il massimo con il minimo sforzo).

Possiamo comunque generalizzare le limitazioni del teorema che sono:

1) - Esso fornisce solo la migliore stima possibile della media e della deviazione standard.

2) - Per campioni di numerosità inferiore a 15 è senz'altro poco accurato, stante la possibilità di trarre casualmente campioni non rappresentativi della vera media.

Ed allora? Un nostro critico lettore potrebbe domandare: Se è del tutto preferibile impiegare il programma per il calcolo della deviazione standard che ci evita, per quanto possibile, errori di valutazione, a cosa mai può servire un programma che, per definizione, ci porta ad assumere rischi?

Intanto per il costo, sappiamo che la molla ultima che spinge i componenti umani è costituita dal prezzo che si è disposti a pagare per ottenere certi risultati; se ci è data la possibilità di ottenere certi risultati con un modello che, posto un certo rischio fissato a priori, impiega meno energie per arrivare ad una certa

conoscenza, ebbene, nessuno può pretendere di più.
Qualsiasi produttore, anche la Texas che ha fabbricato questo computer, non può logicamente collaudare ogni pezzo prodotto, sarebbe del tutto antieconomico e, persino, inutile.
Si organizza pertanto un collaudo statistico che, fissato il rischio del produttore, chiamiamolo α e il rischio del compratore, β, permette attraverso opportune procedure, dette piani di campionamento, di accettare o rifiutare lotti di produzione. E questo, naturalmente con il massimo impiego delle risorse statistiche e con il minimo costo.
Credo, ma non ne sono sicuro, che questo tipo di metodologia chiamata piani di campionamento, sia stata fissata dall'esercito americano durante la guerra di Corea per conciliare le esigenze del Governo e quelle dei produttori e sono regolate come U.S. Military Standard.
Il problema, al solito, è che il rischio α del produttore di scartare come difettoso un lotto in realtà buono, guarda caso, è stato regolato dal produttore stesso mentre il rischio β, o rischio del consumatore di accettare un lotto non conforme alle specifiche è fissato automaticamente dal produttore in funzione di α.
Ma questa faccenda dei piani di campionamento è un'altra storia e ne parleremo in un'altra puntata.

PAOLO CIVARDI

MASTER CONTROL PROGRAM

Master Control Program è uno di quei programmi di archiviazione, o data base, dove è possibile conservare su dispositivo magnetico (in questo caso una cassetta) una serie di dati con la possibilità di recuperarli e gestirli in modi differenti.
Esaminiamo ora, in particolare, le funzioni di questo archivio di programmi. Dopo lo schermo di presentazione, il computer vi chiederà, su sfondo nero e in caratteri verdi, quasi a simulare un monitor, un numero di codice che voi fisserete. Questo numero (lungo al massimo 93 caratteri) rappresenta la vostra personale chiave d'accesso e dovrà essere mantenuto per tutta la durata del programma.
Ad operazione conclusa, apparirà ai vostri occhi lo schermo principale con le tre opzioni di fine lavoro, creazione schedario e aggiornamento (fig. 1). In caso di fine lavoro, e in presenza di dati, il computer entrerà nella fase di registrazione di questi sul dispositivo CS1. Chi, a questo proposito, volesse avvalersi del secondo registratore non dovrà fare altro che sostituire nella linea 580 «CS1» con «CS2».
Se, premendo il tasto uno, sceglierete di creare ex novo un archivio dovrete introdurre i dati, uno alla volta, come segue: il primo dato richiesto, nel caso che si imposti il primo record, sarà il numero di codice e il nome dell'archivio, altrimenti si passerà direttamente al nome del programma che, dal momento che è contenuto in una variabile di stringa, può essere contemporaneamente numerico e alfanumerico.
Una volta premuto l'*ENTER* per confermare l'avvenuta operazione di scrittura, dovrete introdurre il genere del programma, che per comodità è stato diviso in tre gruppi: giochi, utilità-matematica, altro. Ciascun genere è stato poi identificato con un colore che, nell'ordine, risulta essere il giallo, il blu e il verde. Per ottenere questa etichetta, che non deve essere né inferiore e né superiore ai tre caratteri, basta premere contemporaneamente i tasti di

CONTROL e *Q* per i giochi; *CONTROL* e *A* nel caso di programmi scientifici; *CONTROL* e *H* per tutti gli altri. Le rimanenti voci da introdurre saranno il tipo di linguaggio di programmazione usato (basic, extended, ecc., ecc.), il nome dell'autore, l'occupazione totale di memoria (espressa in K), il numero della cassetta su cui è stato registrato il programma, in che lingua (italiano, inglese, ecc.) è stato scritto e, per finire, il costo totale.
Al termine di tutte queste operazioni vi verrà fornita la «scheda» completa del software da voi indicato (fig. 3). Niente paura in caso di errori, potrete infatti, rispondendo N alla domanda «VA BENE?», tornare a correggere partendo dalla penultima domanda e risalendo, semmai, con il tasto di uguale (=) fino alla prima.
Da notare che questa operazione è possibile anche durante l'imissione principale e attuabile in un secondo tempo con l'opzione 2 del sottoprogramma *AGGIORNAMENTO*. La terza ed ultima opzione presuppone l'esistenza di un file (archivio) registrato precedentemente su cassetta.
Una volta caricati tutti i dati in memoria, potrete cominciare le modifiche il cui numero è riassunto in cinque opzioni come da fig. 2 (e che ora esamineremo una alla volta). Il tasto **zero** vi riporta automaticamente al quadro principale; il tasto **uno** procede ad inserire dei programmi aggiuntivi accodandoli a quelli già esistenti; il **due**, come si era detto, serve a modificare una parte o tutto il contenuto di una «scheda» identificata dal suo numero di codice, rintracciabile, per gli smemorati, con l'opzione 4. Il computer provvederà all'eliminazione fisica di una «scheda» una volta premuto il tasto 3. Anche qui prevedendo un vostro possibile errore è stata introdotta una line di «salvataggio» che entrerà in funzione battendo 999 al posto del numero di codice. Con il **quattro** potrete avere una visione completa dei dati (3) oppure fare una ricerca per nome (2) o per codice (1) e tornare poi tranquillamente al quadro principale (0) (fig. 4). Con la quinta opzione avrete un'automatica ordinazione alfabetica dei programmi.

ULTIME CONSIDERAZIONI E CONSIGLI

Il numero massimo di «schede» che l'M.C.P. può gestire è

M.C.P.

Genere
Scritto in Basic
da Squintani Marco
in italiano
occupa 11K
Cassetta 5
Costo L. 10.000
Va bene (S/N)?

Fig. 4

Schedario programmi

0 Fine lavoro
1 Creazione schedario
2 Aggiornamento schedario

Fig. 1

Schedario programmi

0 Quadro principale
1 Inserimento programmi
2 Variazione programmi
3 Eliminazione programmi
4 Visualizzazione programmi
5 Ordinamento alfabetico

Fig. 2

Visualizzazione programmi

0 Quadro principale
1 Ricerca per codice
2 Ricerca per nome
3 Vis. totale

Fig. 3

stato ridotto per ragioni di spazio e memoria a 100, come si può facilmente vedere dalla linea 460, per cui si consiglia di archiviare i vostri programmi raggruppandoli con un certo ordine logico.
Mettete sempre particolare attenzione nell'introduzione dei dati nelle *INPUT*, evitando di chiamare in modo uguale «schede» diverse. La somma della lunghezza dei campi non deve superare i 128 caratteri. La cosa più importante resta comunque quella di non bloccare mai l'esecuzione del programma con un *BREAK POINT*, se non volete perdere irrimediabilmente e definitivamente tutti i vostri dati.
Questo archivio, frutto di tante ore di programmazione, volte a renderlo il più completo e versatile possibile, offre ugualmente lo spunto ai lettori più intraprendenti per sfruttarlo come ottima base ai loro adattamenti e modifiche. Impiegatelo al meglio.

MARCO SQUINTANI

Lista variabili principali

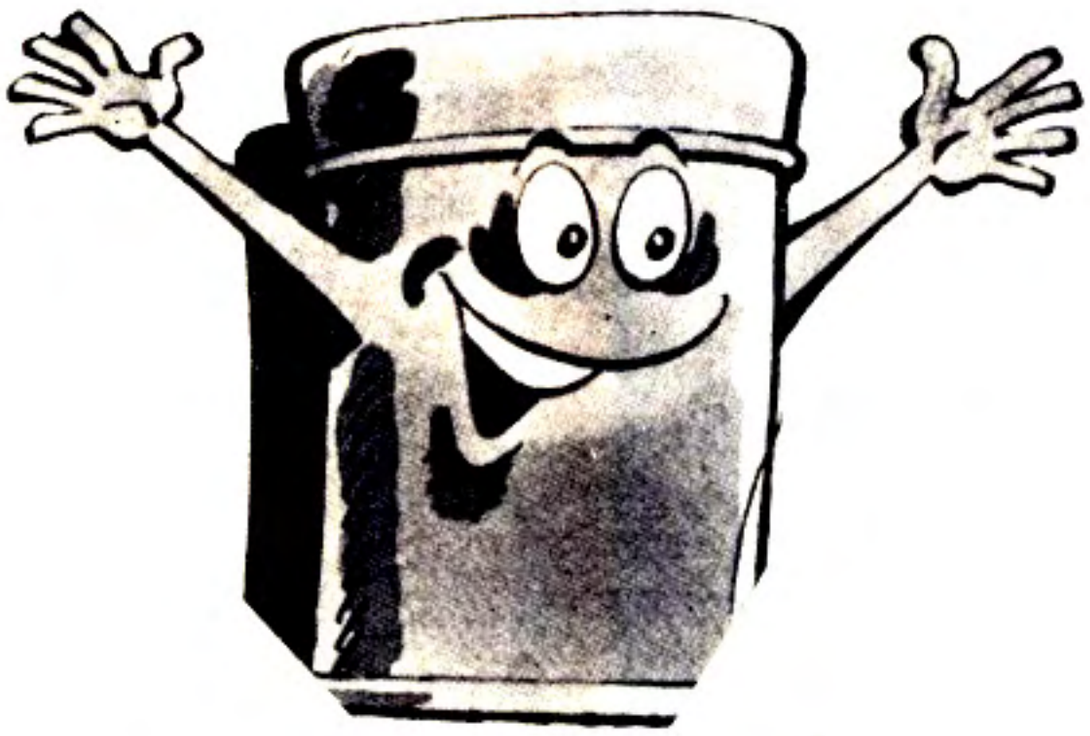
A	=	Tasto premuto nelle CALL KEY
C	=	Numero dei programmi inseriti
1	=	Nome del file
K$	=	Nome dello schedario
F (X)	=	Numero di codice del programma inserito
A$ (X)	=	Campo contenente il nome del programma
B$ (X)	=	Genere
C$ (X)	=	Linguaggio di programmazione usato
D$ (X)	=	Nome dell'autore del programma
E$ (X)	=	Memoria totale occupata (K)
F$ (X)	=	Numero della cassetta
G$ (X)	=	Lingua
H$ (X)	=	Costo totale
IMM	=	Codice immesso
COD	=	Codice reale del file

Lista variabili di ciclo

CH$	=	Stringa contenente la definizione grafica
CC	=	Definizione colori
T	=	Variabile di tempo.

Remarks

100 - 410	Definizione colori e caratteri
420 - 440	Schermo principale
450 - 470	Dimensionamento campi di lavoro
480 - 500	Immissione numero di codice
510 - 550	CALL KEY principale
560 - 670	Scrittura file su cassetta
680 - 910	Lettura file da cassetta
920 - 970	Subroutine di stampa titolo
980 - 1070	Quadro principale
1080 - 1180	Quadro di aggiornamento
1190 - 1870	Procedura di inserimento dei programmi
1880 - 2410	Procedura di variazione
2420 - 2600	Eliminazione programmi
2610 - 2720	Quadro di opzioni di visualizzazione
2730 - 2780	Ricerca per numero di codice
2790 - 2860	Ricerca per nome
2870 - 3070	Visione totale
3080 - 3110	Controllo del numero di codice
3120 - 3130	Subroutine di stampa linea
3140 - 3630	Ordinazione alfabetica
3640 - 3710	Subroutine di presentazione



HOT DOG

Penso cha a tutti voi sia noto il nome BUGER TIME. È uno dei giochi più originali e divertenti degli ultimi anni. Quella che ora la MMG vi presenta è una versione per TI 99 + EXTENDED BASIC.

Ciò che colpisce maggiormente in questo gioco è senz'altro la grafica: molto ben riuscita ed accattivante. Le regole del gioco sono rimaste pressoché quelle originali. Il nostro simpatico cuoco è anche qui alle prese con dei grossi sandwiches, che dovrà formare, fetta per fetta, camminandoci sopra. A complicare le cose troveremo dei simpatici «cattivi». Nella stesura del programma le maggiori difficoltà sono state riscontrate nel verificare le coincidenze tra gli SPRITES, le scalette ed i vari piani. Alla fine il problema è stato risolto memorizzando tutte le posizioni possibili in una matrice, per velocità più elevate di quella usata per il nostro cuoco; l'unico appunto da fare è il notevole dispendio di memoria.

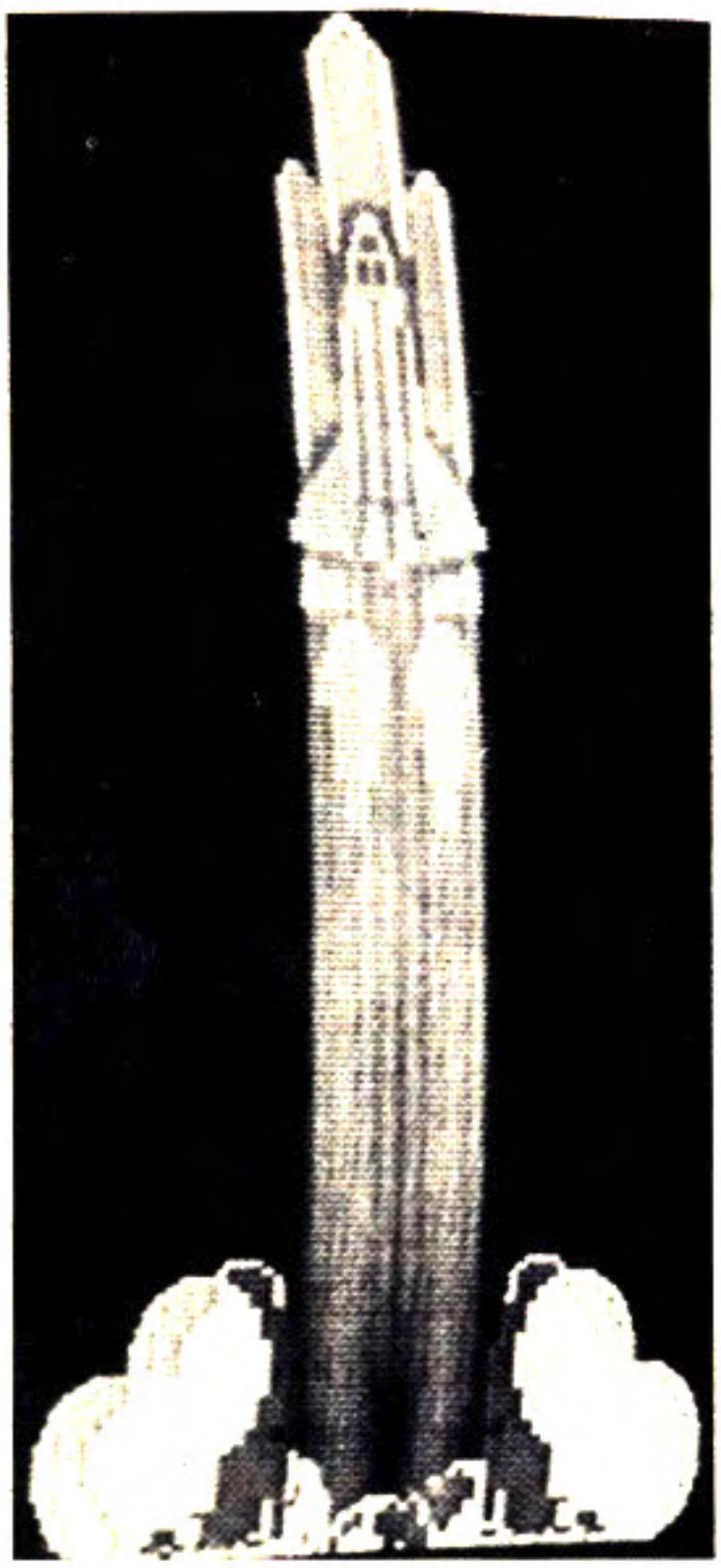
Il movimento degli inseguitori durante il primo quadro è casuale; dal secondo quadro in poi il simpatico cuoco sarà rincorso dagli inseguitori.

Potrà accadervi, qualche volta, di non riuscire a sfuggire ai salsicciotti: allora potete sempre disorientarli «pepandoli», cioè schiacciando la «Q». Purtroppo a volte uno degli inseguitori incomincia a vagare sullo schermo, per poi tornare a fare il proprio dovere. Questo «bug» è stato comunque riscontrato anche sulla cartuccia per TI 99 «burger time», notevole sotto ogni aspetto.

Quindi... detto questo, detto tutto.

Ora tocca a voi, armatevi di pazienza e... BUON APPETITO.

MMG
MAURO MATTIAZZI



STRIKE DEFENCE

ANALISI DEL LISTATO

10 - 100	Rem
110 - 230	Presentazione gioco e impostazione a uno o due giocatori
240 - 630	Definizione caratteri e colori
640 - 780	Disegno generale della scena
790 - 810	Disegna l'aereo
820	Controllo uno/due giocatori
830 - 960	Coordinate per movimento casuale aereo
970 - 1010	Controlla esattezza delle coordinate
1020 - 1030	Controlla se l'aereo urta il missile
1040 - 1050	Movimento aereo
1060 - 1340	Sgancio e movimento bomba. Controllo urti
1350 - 1650	Comandi e coordinate per movimento missile
1660 - 1790	Lancio missile
1800 - 2380	Coordinate e movimento missile. Controlli
2390 - 2450	Esplosione in volo del missile
2460 - 2470	Controllo urti missile
2480 - 2560	Esplosione aereo che urta il missile e n° aerei persi
2570 - 2630	Esplosione missile che urta aereo e bomba
2640 - 2660	Azzeramento variabili bomba
2670 - 2680	Numero aerei colpiti
2690 - 2730	Esplosione bomba che urta
2740 - 2770	Azzeramento variabili missile
2780 - 2800	Controllo serbatoi colpiti
2810 - 2950	Fine del gioco e punteggi
2960 - 3060	Comandi aereo per il secondo giocatore
3070	Fine listato

Scopo del gioco è quello di riuscire ad abbattere il più alto numero di aerei nemici, prima che questi distruggano il Vostro amato deposito di carburante nucleare, formato da cinque grossi serbatoi protetti da uno strato di cemento e nascosti tra gli alberi.

Per riuscire in questo difficile compito avete a disposizione una efficiente rampa lanciamissili, e dei missili teleguidati che dirigerete contro gli aerei nemici.

Il gioco è per due giocatori o per uno. In due, il secondo giocatore guida l'aereo e sgancia le bombe.

Ma vediamo cosa succede una volta dato il RUN.

Dopo la presentazione, il computer vi chiede se volete giocare contro di lui. In caso di risposta negativa, potete giocare in due.

Il giocatore che comanda l'aereo ha a sua disposizione i seguenti tasti: *E* per alzarsi di quota, *X* per abbassarsi, *D* per sganciare la bomba. Sgancita la bomba, bisogna attendere che essa esploda prima di poterne lanciare un'altra.
Il giocatore lanciamissili, invece, lancia il suo missile con la *L* e lo guida con i tasti *UIOJKNM* e, in direzioni chiaramente intuibili. Bisogna attendere che il missile sia esploso prima di poterne lanciare un altro.
La quota minima di volo è proprio sopra gli alberi: l'aereo non può abbassarsi ulteriormente, ed il missile che cerchi di farlo esploderebbe subito, così come esploderebbe se cercasse di superare lo schermo. L'aereo, invece, che si muove automaticamente sempre in avanti, raggiunta la fine dello schermo, riappare sull'altro lato, pronto a un nuovo attacco. Bisogna tenere presente che la gittata del missile è limitata quindi attenti ...
Il gioco finisce quando il deposito è stato distrutto. Il giocatore lanciamissili vince qualora abbia distrutto più di 9 aerei nemici.
Volendo si possono inserire queste linee:
2525 If Aereo = 1Ø then 281Ø
2675 If Aereo = 1Ø then 281Ø
con le quali il gioco terminerebbe quando il giocatore lanciamissili ha ormai vinto.
Ancora, agendo sulla linea 28ØØ si possono diminuire il numero di depositi da colpire, e sulla linea 238Ø si può variare la gittata del missile.
Infine scegliendo di giocare contro il computer, voi dovrete solo occuparvi del missile: al resto penserà lui, e vedrete che saprà darvi filo da torcere.

JOHN FRANK BATTISTA

LISTA VARIABILI

MUS	Variabile per frequenza musica
RIS $	Risposta domanda uno/due giocatori
GIOC	Numero giocatori
TER $	Codice carattere terreno/copertura
EN $	Depositi energia (codice carattere)
ALB 1/2 $	Codice caratteri alberi
RAMPAS $	Codice carattere rampa del missile
MISS I/8 $	Codice caratteri del missile per le otto direzioni
AEREO $	Codice carattere aereo
ESPLOSIONE $	Codice carattere esplosioni
BOMBA $	Codice carattere bomba
TERRA	Coordinate orrizzontali del terreno
ALB	coordinate verticali degli alberi
CAO/CAV	Coordinate aereo
MA	Direzione aereo
OA	Controlla cosa urta l'aereo
MSP	Segnale di bomba sganciata
COB/CVB	Coordinate bomba
DLBG	Decisione di lancio della bomba da parte del giocatore due
SP	Decisione casuale sgancio bomba
OSB	Cosa urta la bomba se viene sganciata
K,S	Call key giocatore I (il lanciamissili)
DIR	Direzione missile
SPM	Segnale di missile sparato
LMO/LMV	Coordinate del missile
OLM	Cosa urta il missile
LM	Conta i lanci di missile (utile per modifiche al programma)
FM	Dà la forma esatta al missile a seconda della sua direzione
MOVM	Conta i movimenti del missile (gittata)
AEREO	Conta gli aerei distrutti
OB	Cosa urta la bomba mentre cade
ENC	Conta i depositi di energia distrutti
CC	Set caratteri per Call Color
DELAY	Ritardo musicale
GK, GS	Call key giocatore 2 (aereo)

FAMILY SPREADSHEET

Questo programma come dice il nome stesso è un tabellone elettronico orientato verso la risoluzione dei problemi della contabilità familiare, cioè vi permette di ordinare, mese per mese, le entrate e le uscite.
Appena dato il RUN occorre aspettare qualche istante per fare in modo che il vostro TI 99 abbia il tempo di definire i nuovi caratteri minuscoli, dopo di che vi verrà presentato il menù principale. Se è la prima volta che utilizzate questo programma dovrete dire al computer quali sono le voci del vostro bilancio, per fare ciò basta nel menù l'opzione di creazione delle voci. Dopo che avrete introdotto fino a un massimo di 7 voci il computer le registrerà su nastro per cui le volte successive sarà sufficiente introdurre tale cassetta su richiesta del computer, che provvederà a caricarle in memoria.
Per inserire i dati veri e propri occorre scegliere nel menù l'opzione di modifica, quindi vi verrà richiesto il quadrimestre che vi interessa oppure se volete vedere il bilancio annuale il quale a sua volta vi chiederà se volete anche un istogramma del bilancio annuale.
Se invece avete inserito il numero del quadrimestre che vi interessa vi apparirà un tabellone che conterrà i dati relativi a quel quadrimestre, se non li avete in precedenza caricati da cassetta con l'apposita opzione del menù, vi saranno tutti zeri. A questo punto avete diverse possibilità di scelta: se premete FCTN 9 tornate alla scelta del quadrimestre mentre invece premendo un tasto qualunque, diverso però dall'ENTER, tornate al menù. Premendo proprio l'ENTER avete la possibilità di inserire i vostri dati. Appena premuto ENTER comparirà una freccia che potete spostare sul mese che vi interessa con i tasti -S- e -D-. Ora avete di nuovo due possibilità di scelta: se dovete inserire i dati di tutto il mese basterà premere una seconda volta ENTER e inserire i dati uno dopo l'altro; se mentre state digitando i dati volete uscire basterà inserire il simbolo della chiocciolina ∂ al posto del numero, tornerete così alla precedente scelta delle funzioni.
Se dovete fare soltanto delle modifiche, dopo aver portato la prima freccia sul mese desiderato basterà premere il tasto -X-: apparirà così una seconda freccia che, usando i tasti -E- e -X-, potrete portare sul dato da modificare, a questo punto premendo ENTER potrete effettuare la modifica.
Tutti questi procedimenti sono piuttosto complicati da descrivere a parole, comunque vedrete che una volta compreso il meccanismo di funzionamento i comandi risultano semplici e intuitivi. Comunque per ogni evenienza nel programma sono com-

prese le istruzioni che possono essere richiamate dal menù principale. Dopo aver introdotto dei nuovi dati ricordatevi di salvarli su nastro altrimenti se fermate il programma essi andranno persi, per questa ragione il programma è protetto da eventuali errori che possono succedere, ad esempio nelle operazioni con il registratore a cassette.
Se avete la stampante e desiderate un tabulato del vostro bilancio potete usare l'opzione 6 del menù per avere un'uscita su stampante.

GIANPAOLO BOTTIN

PASSIAMO ORA AD ANALIZZARE LE PRINCIPALI RUOTINE DEL PROGRAMMA

100 - 200	Inizializzazione variabili e colori
210 - 260	Visualizzazione menù e scelta funzione
270 - 280	Fine
290 - 420	Istruzioni
430 - 640	Visualizzazione e modifica dati
650 - 760	Lettura dati dal registratore a cassette
770 - 890	Scrittura dati su file
900 - 1090	Sub visualizzazione tabella
1100 - 1200	Sub modifica dati
1210 - 1520	Sub caratteri minuscoli
1530 - 1560	Controllo dell'errore
1570 - 1590	Lettura voci del bilancio
1600 - 1640	Creazione e registrazione delle voci
1650 - 1670	Avvertimento che si hanno già dei dati in memoria
1680	Errore in caso di memoria vuota
1690 - 1740	Sub visualizzazione bilancio annuale
1750 - 1820	Sub traccia istogrammi
1830 - 1950	Sub per uscita su stampante

PRINCIPALI VARIABILI USATE:

∂(12,8)	Matrice contenente i dati relativi alle entrate e alle uscite
](12)	Matrice contenente le uscite totali
-(12)	Matrice contenente i risparmi mese per mese
M$(3)	Contiene i nomi dei mesi
V∂$(7)	Contiene i nomi delle voci del bilancio
T-,T],HJ	Flags per evitare operazioni inopportune
	K,S,W,WW,
	X,Y,MS,
	Q,[]$,NM,
G,H,H1	transistori usate per cicli FOR - NEXT, CALL KEY ecc.

Avete sviluppato un programma originale, divertente, utile? Volete vederlo pubblicato su uno dei prossimi numeri di

TI-99 NEWSOFT?

Leggete allora con attenzione *a pag. 31-32* *e mandateci le vostre proposte*

MISSILE

COMMAND

Questo programma è la versione per TI 99 dell'omonimo famoso gioco da arcade. Sono richiesti il modulo EXTENDED BASIC e i JOYSTICKS; non si consiglia nessun adattamento alla tastiera, poichè sono molto importanti le diagonali e, anche se si abilitassero i tasti-cursore e W, Z, C, R la manovra risulterebbe così complicata da perdere inevitabilmente in giocabilità.

IL GIOCO

Dopo aver disinserito il tasto ALPHA LOCK e dato il RUN, appare la presentazione, il computer quindi vi chiederà di premere un tasto: ecco che inizia il gioco.

Sullo schermo, in alto, potrete trovare quale delle fortezze da difendere sarà il bersaglio del missile (o meteorite?! Fate voi). Dovrete quindi piazzare il vostro mirino su di esso e premere il pulsante di fuoco: se avrete buona mira, partirà un missile controffensivo dal cannoncino al centro, che colpirà inesorabilmente il nemico. Inutile dire che, se il missile avversario raggiungerà la fortezza, questa si disintegrerà. Il gioco finisce quando tutte le città sono state distrutte oppure quando vi beccano il cannoncino (il computer avvisa l'attacco fatidico stampando il carattere nemico al posto del numero della FORTEZZA MINACCIATA).

Ogni volta che colpirete il bersaglio, il vostro mirino si troverà sempre più in alto, cosicchè il gioco risulterà più impegnativo.
Riguardo il gioco non c'è più niente da dire: passiamo dunque alle caratteristiche tecniche del programma.

CARATTERISTICHE TECNICHE

Il mio proposito principale era quello di fare un gioco VELOCE e con COINCIDENZE PERFETTE !!!. Chi ha l'EXTENDED sa bene cosa voglio dire e come sia difficile realizzare questo: difficile, ma non impossibile.
Ho pensato quindi di controllare la coincidenza tra missile

e mirino soltanto quando si preme il pulsante di fuoco; ed invece della CALL COINC, ho usato una CALL PEEK che ha la stessa funzione: il risultato, a parer mio, è ottimo. Altra caratteristica del programma è la routine dalla linea 390 alla 430, che determina quale delle fortezze deve essere minacciata: da notare che non capiterà mai un missile che andrà a vuoto o verrà distrutto dal vostro valoroso intervento o distruggerà la città scelta.
Ora non vi resta che caricare il gioco e difendere la terra dagli attacchi alieni!

CARLO LOVADINA

ANALISI DEL LISTATO

100 - 160	REM introduttive
170 - 280	Definizione variabili e caratteri
290 - 380	Definizione dello schermo
390 - 430	Routine di scelta bersaglio
440 - 470	Sprite: definizione
480 - 540	LOOP PRINCIPALE
550 - 710	Routine di cancellazione città distrutta
720 - 780	Schermata finale
790 - 870	Routine di determinazione ascissa bersaglio
880 - 930	Routine di lancio missile controffensivo
940 - 975	Routine esplosione missile nemico
980 - 1030	Routine di distruzione città colpita
1040 - 1320	presentazione
1330 - 1370	Routine di sparo
1400 - 1440	Musica d'introduzione

LISTA DELLE PRINCIPALI VARIABILI

PU	Punti totalizzati
CI (QUALE)	Variabile che si occupa dello stato delle città (distrutta o meno)
QUALE	Variabile di scelta bersaglio
DIFF	Determina la difficoltà del gioco
M	Numero di SPRITE in movimento
HI	Punteggio massimo
BI	Ascisse delle città bersaglio
N	Variabile di ritorno che controlla la coincidenza

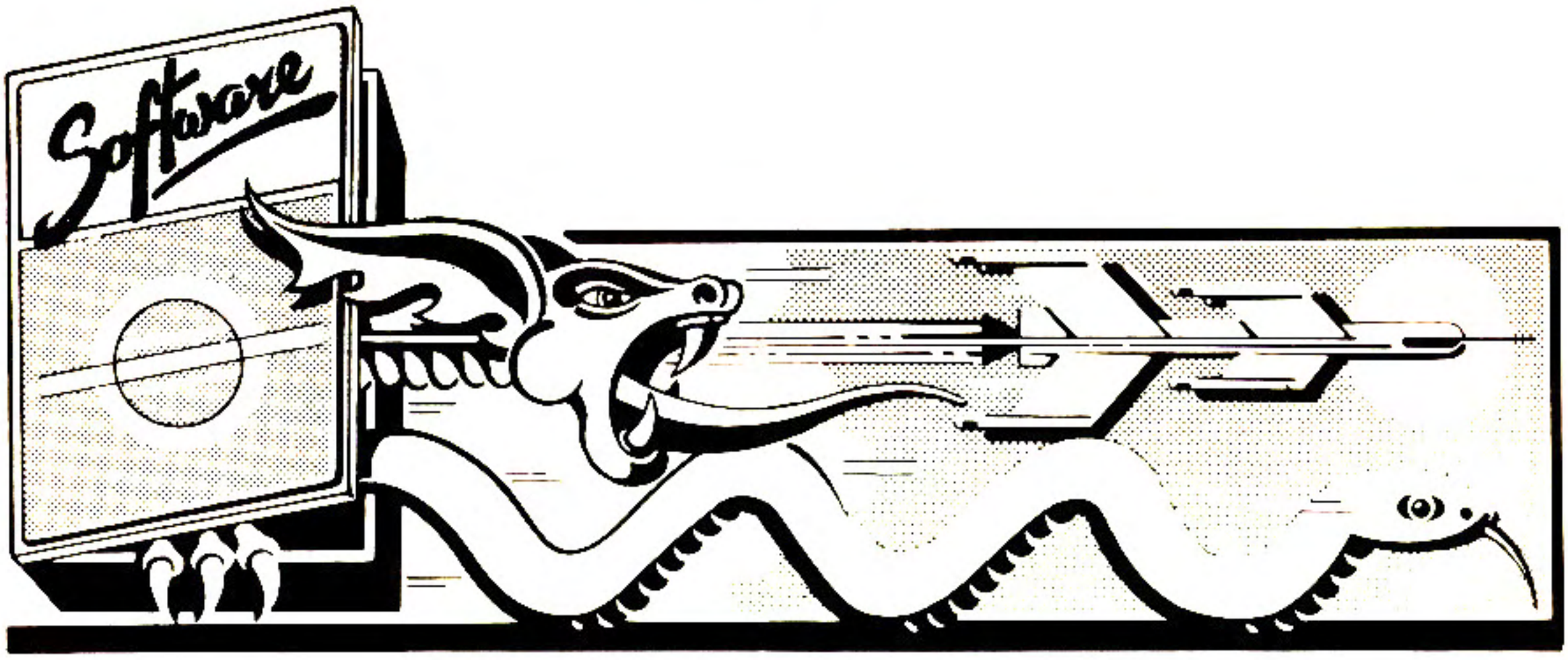
Avete acquistato il Numero di Marzo?

Ma sapete che sono usciti i Numeri di Giugno Settembre Ottobre, Novembre, Dicembre e Gennaio Febbraio?

Per riceverli direttamente a casa vostra, scriveteci! Ve li invieremo in contrassegno a L. 15000 (+spese cadauno).

TI 99 NEWSOFT
Via S. Jacini, 4
20121 MILANO

NIBBLER



Questo programma è la traduzione in TI BASIC di un programma già pubblicato nel primo numero di TI 99 NEWSOFT ma che era scritto per l'EXTENDED BASIC. Ora abbiamo pensato di proporre questo gioco anche per i possessori di TI 99 sprovvisti di questo modulo.
Nonostante il gioco sia lo stesso, il programma si differenzia notevolmente da quello in EXTENDED BASIC poichè il TI Basic è molto meno potente, come linguaggio di programmazione, rispetto all'Extended Basic.
Per chi non conoscesse ancora la dinamica del gioco ne daremo ora una breve descrizione. Voi impersonate un serpente imprigionato in un labirinto in cui vi sono delle prede, dovete mangiare tutte le prede per passare allo schermo successivo, facendo però attenzione a non mangiarvi la coda. Per muovere il Nibbler potete usare il Joystick oppure i tasti con le frecce, E,S,D,X. Attenzione però ogni volta che mangiate una preda il Nibbler si allunga rendendo più difficile il compito di mangiare le altre.

MAURO MATTIAZZI

Descrizione del programma:

100 - 360:	*inizializzazione caratteri, colori e variabili*
370 - 810:	*stampa labirinto e inizializzazione variabili*
820:	*inizio ciclo principale*
820 - 1070:	*lettura joystick o tastiera e movimento testa del serpente*
1080 - 1110:	*se è stata mangiata una preda fa allungare il serpente*
1120 - 1160:	*altrimenti cancella un pezzo di coda*
1170 - 1370:	*routine per il movimento automatico in caso di ostacolo*
1380 - 1490:	*cancella serpente stampa punteggio*
1500 - 1660:	*passaggio allo schermo successivo*
1670 - 1910:	*routine di Game Over*
1920 - 2400:	*presentazione*
2410 - 2470:	*data per il labirinto*
2480 - 2570:	*sub Display at*

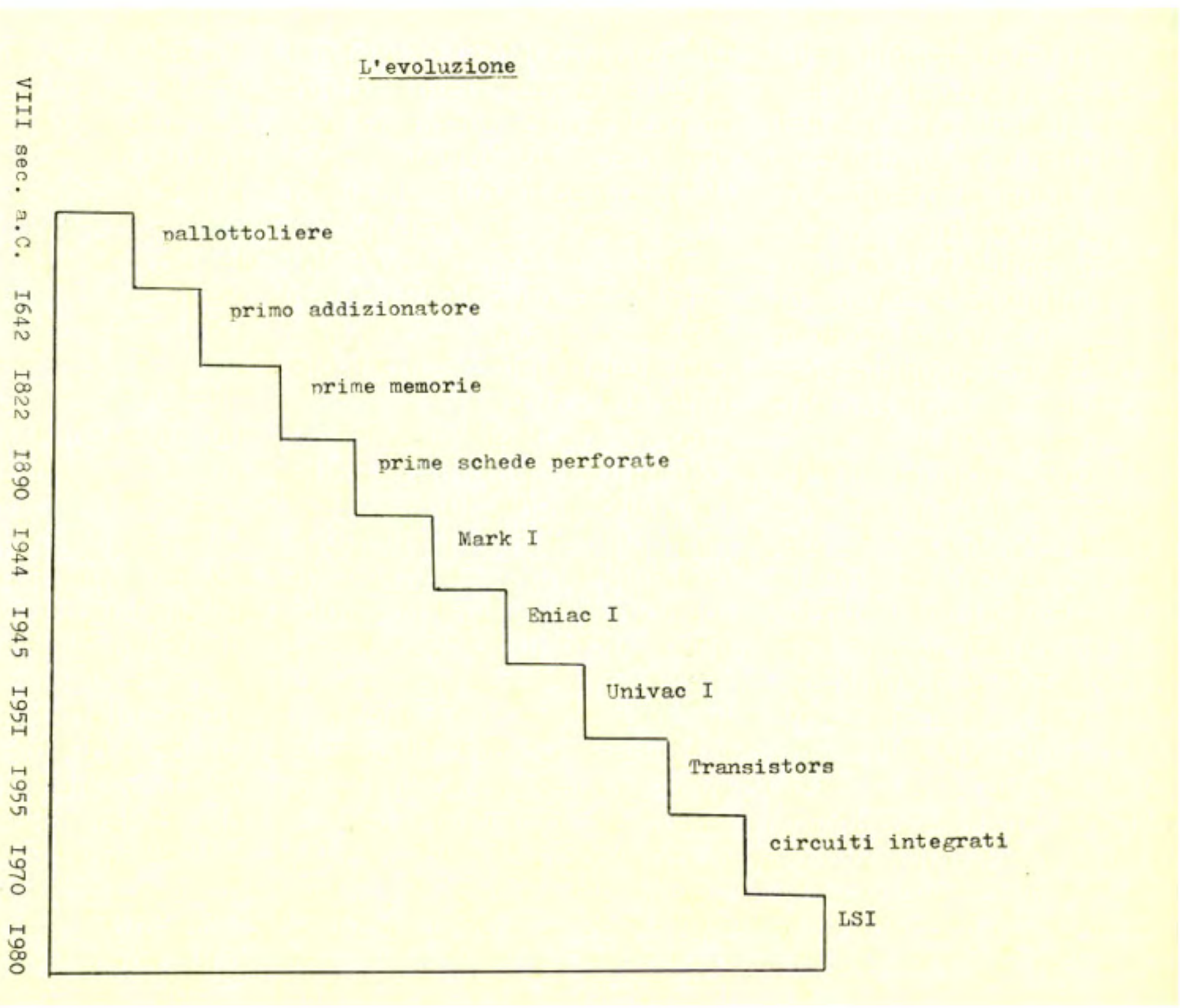
IL COMPUTER, QUESTO SCONOSCIUTO...

Nato per rispondere a problemi algebrici sempre più complessi e per supplire alla scarsa velocità di calcolo della mente umana, il computer è oggi entrato a far parte della vita di tutti i giorni con estrema disinvoltura e accativante simpatia. Non più fantastica chimera risevata ai grandi scienziati o a facoltosi industriali, ma tangibile presenza alla portata (anche economica) di tutti.
Quanti però conoscono le sue lontane e travagliate evoluzioni che l'hanno condotto attraverso i secoli all'attuale aspetto? Le prime testimonianze di uno strumento costruito allo scopo di facilitare e velocizzare i processi di calcolo fanno capo alla Cina dell'VIII secolo a. C., dove troviamo il familiare pallottoliere che ancor oggi, nonostante la sua veneranda età qualcuno usa, riuscendo persino a estrarre le radici quadrate.

Più avanti nei secoli, poiché la cultura e quindi la tecnologia avevano il loro centro nel Mediterraneo, venne introdotta da parte dei romani una variante del pallottoliere: l'abaco, o abacus che dir si voglia. Bisognerà però aspettare fino al 1623, con Wilhelm Schickard, e poi il 1942, con Blaise Pescal, per vedere sviluppare un sistema di calcolo meccanizzato.
Anche se in seguito venne introdotto l'operatore della moltiplicazione, tutte queste macchine sono state per lungo tempo considerate soltanto come curiosità scientifiche e non utilizzate a scopi pratici.
Intorno al 1822 un matematico inglese, Charles Babbage, diede forma al progetto di una macchina con memorie e con possibilità di svolgere operazioni multiple; e gran parte del lavoro teorico da lui svolto rappresenta tuttora la base della moderna informatica.
Nel 1890 il governo degli Stati Uniti, culla della maggior parte dei computer presenti sul mercato oggi, si trovò di fronte al grave problema dell'archiviazione e dell'utilizzazione di una quantità notevole di dati prevenienti dal decennale censimento della popolazione. A questo proposito, Hermann Halleith realizzò un rivoluzionario sistema di archiviazione meccanizzato a manovella, basato sulla perforazione di schede. Una piccola curiosità: questo signore, per sfruttare al meglio la sua invenzione, fondò una ditta che in pochi anni si ingrandì fino a diventare l'attuale colosso quale è l'International Business Machines, meglio conosciuta con la sigla IBM; niente male, eh?
Nel nostro lungo cammino troviamo ora i primi calcolatori elettromeccanici alimentati da un motore elettrico cui seguono, verso la fine degli

anni 30, macchine a relè elettrici funzionanti secondo le teorie di Babbage.
L'apice del calcolatore elettrico si ebbe però nel 1944 con la costruzione, presso l'università di Harvard (USA), dal Mark 1 che aveva una capacità di calcolo di dieci operazione al secondo, effettuabili in sequenza: ciò significa che era nato il primo esempio di programma.
Sulla spinta dei grossi finanziamenti del periodo bellico, fu realizzato nel dicembre del 1945 l'Eniac, cervello di ventimila valvole termoioniche, mille volte più celere del Mark 1. Era tuttavia necessario, riprogrammarlo, modificando le varie connessioni tra le sue numerose valvole con continue saldature dissaldature; oltre a ciò, sviluppava, dato il grande consumo di energia elettrica, un'enorme quantità di calore che causava il continuo «saltare» delle valvole.
Risolti in parte questi inconvenienti, si potè procedere, nel 1951, alla vendita dei computer da parte della Remington Rand corporation. Il suo modello, l'Univac 1, necessitava comunque di continue manutenzioni. Ma anche quando vennero introdotte le memorie a nuclei magnetici e i transistors, eliminando così i problemi legati alla semplicità d'uso e all'affidabilità, rimaneva sempre l'alto costo della macchina, con conseguente scarsa diffusione delle stesse.
La grande rivoluzione si sca-tenò allorché si introdussero le tecnologie LSI (integrazione su vasta scala) con le quali fu possibile, al costo di pochi dollari, creare microscopici circuiti che, posti su un'unica piastrina di silicio (chip), equivalevano a migliaia di transistors. Forti di queste nuove tecniche, computer poterono finalmente dilagare, alimentando un giro d'affari di proporzioni spaventose.
Pensate, dunque, quanta fatica è occorsa per creare questi piccoli "mostri", prima di tirargli una martellata stremati dai continui e assillanti INCORRECT STATEMENT.

MARCO SQUINTANI



DECATHLON

Programma: «**Decathlon**» della Pewterware.
Configurazione richiesta: **consolle + registratore + cavo + joysticks (opzionali).**

IL GIOCO

Il tema proposto è quello della competizione a livello internazionale nei grandi giochi olimpici. Infatti, a questo gioco vi possono prendere parte da uno a otto giocatori, che si dovranno cimentare in una gara a tempo nelle dieci principali discipline olimpiche: 100 m. piani, salto in lungo, lancio del peso, salto in alto, 400 m. piani, 100 m. ostacoli, lancio del disco, salto con l'asta, lancio del giavellotto e, per ultima, la prova dei 1500 m. Per ciascuna di queste gare i giocatori (o il giocatore) dovranno premere la barra spaziatrice o il tasto di fuoco del joystick, contando il tempo richiesto (da 1 a 10 secondi) immediatamente dopo un segnale sonoro.
La precisione dell'orologio del computer ha uno scarto di 1/5 al secondo, abbastanza vicino perciò alla realtà. Più si dà il tempo vicino a quello corretto, migliore sarà il risultato conseguito nella gara. Le prestazioni ottenute, il piazzamento conseguito e la classifica generale saranno costantemente aggiornati al termine di ogni prova. La grafica, pur nella sua semplicità, risulta ben curata e rende abbastanza fedelmente l'idea del movimento dell'atleta.
Il programma è, una volta tanto, fornito di chiare istruzioni in italiano, sia per il caricamento che per il gioco vero e proprio. Una tabella dei migliori risultati ottenuti vi offrirà un valido termine di paragone. Chi riuscirà a raggiungere il punteggio massimo di 1200 punti risulterà davvero bravo.
Un consiglio per spuntarla sugli amici: allenatevi con un buon orologio al quarzo.

MARCO SQUINTANI

ABBONATEVI
A TI 99
NEWSOFT.
CON SOLE
L. 95.000
RICEVERETE
11 NUMERI
DELLA RIVISTA
E IN PIU'
AVRETE
DIRITTO AGLI
SCONTI SUL
SOFTWARE
E
HARDWARE
OFFERTO DA
TI 99 MARKET

Per abbonarsi
basta inviare
un assegno
non trasferibile
o un vaglia
postale a:

NEWSOFT SNC
Via S. Jacini, 4
20121 MILANO

Spett.le Redazione,
dalla pubblicità fatta, su un mensile specializzato, della Vs. rivista, ho appreso l'esistenza di TI-99 NEWSOFT precipitandomi subito ad ordinarne una copia al mio edicolante, il quale non era al corrente della pubblicazione della Vs. rivista.
La trovo eccellente e, spero che continuerete sulla stessa strada continuando a proporci interessanti offerte commerciali.
Nell'articolo, pubblicato sul n. 4, relativo al MEMORY FULL non menzionate l'eventualità che tale messaggio venga evidenziato anche e soprattutto quando si tenta di caricare da registratore un programma tenendo il peripheral con espansione e drive collegati.
Purtroppo non sono ancora riuscito a risolvere tale dilemma e sono rammaricato dal fatto che non riesca a ricevere su disco tutti i programmi che ho registrato su cassetta.
Ho fatto dunque male a comprare il disk-drive?
Attendo con ansia la soluzione che mi proporrete e con la occasione Vi porgo i miei saluti ed auguri di un proficuo 1985.

MARCELLO FASAMO
Salerno

Il problema di caricamento dei dati, da registratore, con il box di espansione collegato, è provocato dal fatto che l'espansione di memoria, riserva alcuni Kbytes di memoria per poter gestire i files su disco.
Generalmente il numero di files gestiti in difetto è di 3.
Tale numero è tuttavia modificabile con il comando CALL FILES (n. dei files da aprire) per maggiori spiegazioni vedi manuale per l'uso del sistema a dischi.
Se poniamo il numero di files uguale a 1, possiamo impiegare più memoria.
Prima di caricare i programmi che danno il segnale di errore, esegua le seguenti operazioni in modo comando.

CALL FILES (1)
NEW

Dopo riprovi a caricarli.
Il problema dovrebbe essere risolto.

E.M.

Gentilissimi Redattori,
questa lettera è rivolta in particolare ai collaboratori della redazione del NEW SOFT.
Mi piacerebbe che voi dedicaste al TI-99/4A un video game veramente speciale; per fare contento non solo me ma tutti i ragazzi e gli appassionati che hanno la fortuna di avere il TI-99/4A. Ve lo chiedo anche per far vedere il grado di elasticità di un home computer come questo; in particolare mi riferisco a programmi che facciano uso sia di linguaggio assembler e di basic con relativi agganci dall'uno all'altro.

FABRIZIO FANTUZZI
Roma

Fabrizio, come tanti lettori della nostra rivista, ci chiede di pubblicare programmi di alto livello che alla qualità del gioco abbinino routine in linguaggio macchina capaci di sfruttare appieno le capacità del TI 99.
Dobbiamo però naturalmente tenere presente che giochi più complessi e veloci, gestiti dall'Assembler, richiedono configurazioni del computer abbastanza evolute e non tutti i nostri lettori sarebbero poi in grado di usufruire di questi programmi. Cerchiamo quindi di accontentare tutti, pubblicando programmi differenziati, per moduli diversi, anche se naturalmente, il nostro obiettivo resta comunque quello di pubblicare programmi sempre più belli.

E.M.

TI99 NEWSOFT

APRILE 1985 L. 10.000

NUMERO 8

Come è fatto il MODULO EXTENDED BASIC

Impariamo a costruire UN ARCADE GAME: i comandi

IN SOFTWARE & SOFTWARE:
MINICALC, Alta risoluzione,
Robotime, Avventura nel calcolatore

SOMMARIO

□ **TI BASIC**
Vi spieghiamo come usare il cursore — pag. 2

□ **EXTENDED + ESPANSIONE 32 K**
Come è organizzata la memoria. Come è fatto il modulo EXTENDED BASIC — pag. 3

□ **EXTENDED**
Le tecniche per far ruotare i nostri "eroi" — pag. 6

□ **RECENSIONI**
Parliamo della cassetta "UP PERISCOPE" — pag. 7

□ **TI 99/4A**
Le ROM, le PROM e le EPROM. Cosa sono e come lavorano — pag. 8

□ **PROGRAMMAZIONE**
Come costruire un arcade game. Questa volta parliamo dei comandi — pag. 9

□ **MINI MEMORY**
Alta risoluzione per la mini memory del nostro TI 99 — pag. 11

□ **HELPER**
Un programma che aiuterà a tradurre codici binari, decimali, esadecimali — pag. 13

□ **ROBOTIME**
Per diventare dei veri... ROBOT BUSTERS! — pag. 14

□ **UN'AVVENTURA NEL CALCOLATORE**
Una sera, per caso, un viaggio all'interno del computer — pag. 15

□ **INVENZIONI**
Il computer con memoria nascosta — pag. 16

□ **E LE STELLE STANNO A PARLARE**
Il vostro destino nelle mani del computer — pag. 17

□ **DIDATTICA**
I diagrammi a settori circolari — pag. 18

□ **MINI CALC**
Tabellone elettronico in BASIC — pag. 20

□ **LA VIGNETTA** — pag. 26

□ **LA POSTA** — pag. 27

□ **PICCOLI ANNUNCI** — pag. 28

□ **COLLABORATE CON NOI** — pag. 31

□ **TI 99 MARKET** — pag. 33

TI 99 NEWSOFT
ANNO 2 - Numero otto
APRILE 1985
MENSILE
Direttore responsabile:
ANTONIO C. LOSITO
Coordinatore Editoriale:
VINICIO CIVININI
Direttore Amministrativo:
ENRICO CALLERIO
Art Director:
BEPPE RE-FRASCHINI
Illustrazioni:
MARIA TERESA COCOZZA
Segreteria:
CRISTINA FRISON
Redazione:
FIORENZA AURIEMMA
Collaboratori:
SERGIO BORSANI,
GIANPAOLO BOTTIN,
DANIELE CATALFAMO,
PAOLO CIVARDI,
STEFANO DELLI PONTI,
FRANCESCO MACALI,
MAURO MATTIAZZI,
EZIO MONTINI,
LEVIO PEROTTI,
RICCARDO ROSSI,
MARCO SQUINTANI,
ALBERTO STRAFILE,
SEBASTIANO TOMASELLO
TI 99 NEWSOFT è una pubblicazione della **NEWSOFT snc**, via Stefano Jacini 4 - 20121 MI
Tel. 02/807464 - Telex 324284
REGISTRAZIONE AL TRIBUNALE DI MILANO N. 323 DEL 7.7.84

COPYRIGHT NEWSOFT snc
TUTTI I DIRITTI RISERVATI
Manoscritti e foto originali, anche se non pubblicati, non si restituiscono ed è vietata la riproduzione anche parziale di testi, fotografie e listati.

Pubblicità: AV STUDIO
Via Giuseppe Meda, 23
Tel. 02/8357155
Abbonamenti: l'abbonamento annuo a 11 numeri di *TI 99 NEWSOFT* costa L. 95.000 per l'Italia; L. 160.000 per l'Europa (spedizione aerea); L. 200.000 per tutti gli altri paesi.
L'importo, per assegno non trasferibile o per vaglia postale, va inviato a NEWSOFT snc, via S. Jacini 4 - 20121 Milano.
Stampa: CGB, Albegno di Treviolo (BG)
Concessionaria per la distribuzione:
EUROSTAMPA Srl - C.so Vittorio Emanuele 111 - 10123 Torino - Tel. 011/538166-7

L'editore non si assume responsabilità alcuna sul contenuto degli articoli e/o software inviati.
Per gli articoli firmati e/o siglati da collaboratori esterni, la redazione si assume la responsabilità prevista dalle leggi sulla stampa.

Tutti possono collaborare a TI 99 NEWSOFT, sia con articoli che con programmi. Scriveteci o telefonateci in redazione per avere tutti i ragguagli.

GIOCHIAMO CON IL TI 99/4A

Un libro che non deve assolutamente mancare nella vostra libreria.

Potrete riceverlo a casa vostra a sole **L. 9500** + Spese di spedizione.

Per ogni richiesta di contrassegno, utilizzate il tagliando qui sotto spedendolo a:

NEWSOFT snc
via S. Jacini, 4
20121 MILANO

Vogliate spedirmi in contrassegno di L. 9500 + spese il libro: «GIOCHIAMO CON IL TI 99/4A»

Cognome .....

Nome .....

Indirizzo .....

.....

Cap. ..... Loc. .....

I SEGRETI DEL CURSORE

ANCORA TRUCCHI

Il manuale che di solito viene fornito al momento dell'acquisto del computer è uno dei migliori presenti sul mercato: chiaro, semplice, scorrevole, scritto in italiano, forse un po' avaro per l'utente più esperto. Alludiamo alla possibilità di poter fare andare il cursore «al di là del suo naso», intendendo per naso il limite massimo di quattro linee di 28 caratteri l'una.
Ma, più delle parole valgono i fatti; perciò bando alle ciance, e via sulla tastiera. Battete un numero di linea, per esempio 10, e poi un'istruzione basic opportunamente scelta, come può essere la PRINT, seguita da un bel paio di virgolette (FCTN P) e da un messaggio qualsiasi (figura 1). Fate arrivare il cursore fino in fondo, e quando sentirete un cupo suono di richiamo, digitate nuovamente le virgolette e battete ENTER.
Richiamate ora la linea battendo EDIT 10 seguito da ENTER, oppure — più semplicemente — 1O con FCTN E (o FCTN X). Mai, ripetiamo mai, fate 10 ed ENTER perché cancellereste la linea irreparabilmente.
Siete ora con il cursore piazzato sul primo carattere dopo il numero di linea. Una volta passati al di là delle virgolette, date l'ordine di inserimento caratteri (FCTN 2 = INSERT) e scrivete ancora qualcosa. I caratteri si sposteranno di uno spazio a destra ogni volta che batterete un tasto e faranno apparire una linea in più (la quinta). Potrete ottenere ancora una linea ripetendo di nuovo la procedura.
Tutto questo discorso vale ovviamente per quelle altre istruzioni che potrebbero richiedere più di quattro linee (ON GOTO, ON GOSUB, DATA, ecc.).
Attenti a non abusare eccessivamente di questi spazi extra, se non volete far apparire il messaggio LINE TOO LONG.

MARCO SQUINTANI

```
10 PRINT «QUESTA FRASE DEVE OCCUPARE LA BELLEZZA DI QUATTRO LINEE DI TESTO SE VOLETE CHE L'ESPERIMENTO RIESCA».
```

COME È FATTO
IL MODULO EXTENDED BASIC

Il TI 99 ha la caratteristica di cambiare la mappa di memoria ogni volta che un modulo è inserito, a seconda che si tratti di un modulo di gioco o di comando. Il modulo TI EXTENDED BASIC riorganizza tutta la memoria del computer.
Per poter lavorare in linguaggio macchina è assolutamente necessario sapere anche nei più piccoli dettagli, in che modo la memoria è organizzata. Come sappiamo, ci sono quattro tipi di cip di memoria nel nostro computer.
Due di essi servono esclusivamente per farlo funzionare e non c'è modo per accedere ad essi. Il primo tipo è chiamato ROM, l'altro GROM.
Il primo tipo è programmato in linguaggio macchina, mentre il secondo è programmato in uno strano linguaggio chiamato GPL.
Nella maggior parte dei moduli possiamo trovare questi due tipi di cip. Generalmente la ROM è utilizzata per contenere il programma in linguaggio macchina, la GROM per contenere i dati di tipo grafico, ad esempio la definizione dei caratteri ed altro.
Riprenderemo comunque il discorso riguardante le GROM e le ROM nelle prossime puntate.
I rimanenti tipi di cip vengono chiamati rispettivamente RAM e VDP RAM. Nella RAM, possiamo scrivere e leggere liberamente i dati, in essa vengono depositati i programmi basic, e gli eventuali programmi in LM, da noi creati.
Tutte le operazioni riguardanti la gestione degli sprite, colori, caratteri grafici, e tutto quello che riguarda il microprocessore grafico, vengono gestiti da un particolare supporto di memoria appositamente creato per gestire questo tipo di dati, e il suo nome è Video Display Processor RAM.
È fatto divieto all'utente di intervenire in questa particolare area.
Infatti, se non si conosce perfettamente la mappa di tale area, si rischia di mandare facilmente in crash la macchina.
Noi vi forniamo la dettagliata mappa di memoria di tutta la VDP RAM, e con le dovute precauzioni "potrete pasticciare", in quest'area con l'aiuto del programma WRITERLOADER che avete trovato nel numero di marzo.
Passiamo ora alla dettagliata descrizione di ogni singola area di memoria della VDP RAM.
Noterete che i nomi delle varie aree sono in inglese, abbiamo preferito lasciare tale linguaggio per facilitare la ricerca sui vari manuali del microprocessore per esempio manuale EDITOR ASSEMBLER.

SCREEN IMAGE TABLE
dec. 00000-00767
esa.>0000->02FF

In essa sono contenute tutte le locazioni video che possono essere visualizzate e ad ogni casella dello schermo corrisponde un byte. Lo schermo è diviso in 32 colonne e 24 righe, le celle di memoria impiegate sono il risultato di 32*24=768, sottraendo uno perché in ambito LM, viene conteggiato anche zero, e otteniamo che le reali locazioni impiegate sono 767.
Per esempio vogliamo scrivere il carattere M alla riga 12 e alla colonna 16. Dovremo inserire, con il programma WRITERLOADER, alla locazione:
(12-1) * (16-1) = 165, il valore del carattere che vogliamo sia visualizzato.
M=codice ascii (77)+offset (96) = 173.
Ricordatevi di aggiungere sempre l'offset di (96).

SPRITE ATTRIBUTE TABLE
dec. 00768-00879
esa.>0300->036F

In questa area è contenuta una tabella di dati riguardanti gli sprites. Per ogni sprite sono riservati 4 bytes, nei primi due dobbiamo indicare la riga da 0-191 e la colonna da 0-255, mentre nel terzo il carattere con offset 96.
L'ultimo byte è riservato al colore, compreso tra 0 e 15.
Gli sprite che si possono generare sono (112/4)=28.

PATTERN DESCRIPTOR TABLE
dec. 00880-01919
esa. >0370->077F

Il computer deposita tutto il set di caratteri in questa area. Detto set può essere facilmente modificato, cambiando la forma, ad esempio quella del cursore:
Il cursore si trova alla locazione (30*8)+768=1008, ed ha una lunghezza di 8 bytes. Poniamo il caso di voler cambiare la forma del cursore rendendola simile a quella di un PC ibm, allora inseriamo a partire dalla locazione 1008 i valori 0, 0, 0, 0, 0, 0, 0, 225. Il cursore avrà una forma a linetta, molto più simpatica.
A voi altre prove; si intende che per effettuare queste alterazioni è necessario il programma WRITERLOADER.
ATTENZIONE: non toccate ne alterate i caratteri che hanno come codice dei numeri inferiori a 30, perchè sono utilizzati dalla macchina per funzioni interne.

SPRITE MOTION TABLE
dec. 01920-02047
esa. >0780->07FF

Questa parte di memoria è riservata per la velocità degli

ESA.	DEC.	MAPPA V D P RAM DELL' EXTENDED BASIC by E Z I O M O N T I N I
>0000	0	SCREEN IMAGE TABLE contiene tutte le posizioni dei caratteri dello schermo ed ha una lunghezza pari a 32*24=768
>02FF	767	
>0300	768	SPRITE ATTRIBUTE TABLE
>036F	879	
>0370	880	PATTERN DESCRIPTOR TABLE offset da utilizzare +96 (>60) Per trovare l' indirizzo decimale del carattere eseguire la seguente operazione : numero del carattere * 8 + 768 il risultato da l' indirizzo di partenza del carattere .
>077F	1919	
>0780	1920	SPRITE MOTION TABLE
>07FF	2047	
>0800	2048	COLOR TABLE
>081F	2079	
>0820	2080	Con EXP.32K. si hanno le seguenti aree di memoria : PABS STRIGHE TAVOLA DEI SIMBOLI variabili numeriche, numeri di linea, spazio per i programmi.
>36D7	13783	
>35D8	13784	DISK FILE BUFFERS
>3FFF	16383	

	0	1	2	3	4	5	6	7
codice foreground								
codice background								

sprite, è suddivisa in settori di 4 bytes, ogni settore indica la velocità di riga e di colonna di ogni singolo sprite.
Solamente i primi 2 bytes di ogni settore possono essere manipolati, i secondi sono riservati per funzioni di sistema.
Nel primo byte va inserita la velocità di riga, poi nel successivo quella di colonna.
Nei prossimi numeri troverete indicazioni più dettagliate sull'inserimento delle velocità.
Per trovare la velocità dello sprite 10 dovremo eseguire le seguenti operazioni
1920+(10*4)=1960 (per la velocità di riga), 1961 (per quella di colonna).
Sino a quando non vi spiegheremo tutte le modalità per l'inserimento della velocità, limitatevi a compiere delle operazioni di lettura, onde evitare spiacevoli inconvenienti.

COLOR TABLE
dec. 02048-02079
esa. >0800->081F

Questa tavola contiene tutta la tavola dei colori ed è suddivisa in 32 parti, di cui solo 16 sono utilizzabili dall'utente.
Ogni bytes contiene il codice del colore foreground e quello background.
Nei primi 4 bit è presente il codice da >0 a >F fg. e nei secondi il codice da >0 a>F bg.
Per stabilire i colori, scriveteli nel sopraindicato schema in modo binario, poi traduceteli con l'aiuto del programma che vi proponiamo questo mese: HELPER.
Ogni byte di colore definisce il colore di 8 caratteri. Per definire il colore del carattere 128 effettuare le seguenti operazioni:
128/8=16+1=17 il codice appartiene al gruppo 17
2048+17=2065 questa è la locazione dove dovremo inserire il nostro colore, calcolato come abbiamo precedentemente detto.
Le aree di memoria dalla loc. 2080 alla loc. 16383, saranno descritte nel prossimo numero.

EZIO MONTINI

ABBONATEVI
A TI 99
NEWSOFT.
CON SOLE
L. 95.000
RICEVERETE
11 NUMERI
DELLA RIVISTA
E IN PIU'
AVRETE
DIRITTO AGLI
SCONTI SUL
SOFTWARE
E
HARDWARE
OFFERTO DA
TI 99 MARKET

Per abbonarsi
basta inviare
un assegno
non trasferibile
o un vaglia
postale a:

NEWSOFT SNC
Via S. Jacini, 4
20121 MILANO

ROTAZIONE DEGLI SPRITES

Generalmente in giochi come Pac Man, asteroid ed altri ancora, è necessario adottare delle tecniche di animazione per far ruotare il nostro eroe su se stesso.
Per prima cosa dobbiamo definire tanti patterns quante sono le direzioni in cui vogliamo spostarci.
Vi consigliamo di definire i caratteri in modo ordinato, i caratteri devono cioè essere siano definiti in ordine di rotazione oraria e antioraria, perché ciò faciliterà le successive operazioni di gestione dei caratteri. Per ottenere una buona risposta ai comandi di direzione, dobbiamo cercare di comprimere, il più possibile, le routines che svolgono i cambiamenti di direzione e dei conseguenti cambiamenti del pattern.
Vi proponiamo anche questa volta due routines che svolgono tale attività, con una velocità di esecuzione molto più elevata rispetto ai normali metodi usati nella gran parte dei programmi.
Le due routine sono ingloba-te in due piccoli programmi, per farvi meglio capire in che modo possono essere inserite in un eventuale vostro programma.
La routine contenuta nell'esempio numero uno esegue una rotazione di 360 gradi in modo continuo. La seconda, pur basandosi sullo stesso principio di funzionamento, è in grado di fermare la rotazione ad un punto stabilito, consentendo però di ruotare in senso inverso al raggiungimento del limite massimo contrario da noi stabilito.
Entrambi i programmi utilizzano dei metodi di elaborazione logica dei dati numerici. Essi si basano sul funzionamento dei due operatori logici > e < «maggiore e minore». Il loro utilizzo abbrevia molto il tempo di elaborazione dei dati per il computer, permettendoci di sfruttare al meglio le sue possibilità.
Esaminiamo per primo il programma descritto nell'esemprio 2.
I due algoritmi che fanno funzionare il programma sono contenuti rispettivamente alle linee 170 (per il movimento antiorario), 180 (per quello orario).
La variabile P contiene il valore del pattern da utilizzare, che viene incrementato o diminuito a seconda che si voglia girare in senso orario P=P+(P<100)*-1, o orario P=P+(P>96).
Il campo di esistenza dei nostri pattern è compreso tra 96 e 100. Detto numero non sarà mai superato. Per cambiare tale campo di esistenza basta variare il valore contenuto nelle parentesi.
Alla riga 170 è contenuto il valore minimo, alla riga suc-

```
1 !+------------------------+
2 !! Esempio 1 di rotazione!
3 !! continua degli sprites!
4 !! EZIO MONTINI SOFTWARE !
5 !+------------------------+
100 P=96
110 CALL MAGNIFY(2)
120 CALL CHAR(96,"1010101010101010",97,"0102040810204080",98,"00000000FF000000",
99,"8040201008040201",100,"0101010101010101")
130 CALL CLEAR
140 CALL SPRITE(#1,96,2,110,110)
150 CALL KEY(0,K,S)
160 IF S=0 THEN 150
170 IF K=ASC("S")THEN P=P+(P<95):: IF (P<95)=0 THEN P=99
180 IF K=ASC("D")THEN P=P+(P<100)*-1 :: IF (P<100)=0 THEN P=96
190 CALL PATTERN(#1,P)
200 GOTO 150
```

```
1 !+------------------------+
2 !! Esempio 2 di rotazione!
3 !!  controllata  sprites !
4 !! EZIO MONTINI SOFTWARE !
5 !+------------------------+
100 P=96
110 CALL MAGNIFY(2)
120 CALL CHAR(96,"1010101010101010",97,"0102040810204080",98,"00000000FF000000",
99,"8040201008040201",100,"0101010101010101")
130 CALL CLEAR
140 CALL SPRITE(#1,96,2,110,110)
150 CALL KEY(0,K,S)
160 IF S=0 THEN 150
170 IF K=ASC("S")THEN P=P+(P<96)
180 IF K=ASC("D")THEN P=P+(P<100)*-1
190 CALL PATTERN(#1,P)
200 GOTO 150
```

cessiva quello massimo che non deve essere oltrepassato.
Tutto il programma si basa sul principio che, se la condizione posta tra parentesi è verificata, l'operazione logica darà come risultato il valore (-1), in contrario, il valore sarà (0).
Quindi, come nell'esempio, se P=96, (P>96)=0, se P=97, (P>96)=-1 ecc...
Lo stesso principio vale per la riga 170, però il valore di ritorno va moltiplicato per -1, perché dobbiamo incrementare la variabile P.
Il programma dell'esempio 1, pur basandosi sullo stesso principio di quello già descritto, utilizza in modo diverso gli operatori, permettendo una rotazione continua degli sprites. Oltre agli algoritmi già descritti, troviamo altre due espressioni logiche, che all'interno di IF THEN, operano i necessari controlli per impedire di oltrepassare il limite di caratteri consentito.
Alla riga 170 dovremo inserire il valore minimo consentito meno uno, e nella seguente verifica dovremo inserire lo stesso valore, dando a P il valore massimo consentito meno uno.
Alla riga 180, i valori da inserire sono quelli massimi consentiti, senza alcuna sottrazione, ponendo la variabile P al valore minimo consentito.
Tutti questi inserimenti di numeri ci consentiranno di ottenere ed adattare ai vostri programmi questa velocissima routine.

EZIO MONTINI

UP PERISCOPE

Programma: **«UP periscope».**
Configurazione richiesta: **consolle, registratore, cavo modulo extended basic.**

IL GIOCO

Immaginate di essere nei panni di un valente comandante di sottomarino in agguato sul fondo dell'oceano. Vostro compito sarà quello di affondare i carichi destinati al nemico, evitando, nel contempo, di essere a vostra volta affondati.
A disposizione 12 o 24 siluri, come avrete deciso prima di cominciare la caccia, da indirizzare verso il cargo avversario. La navigazione sarà suscettibile di cambiamento, premendo vari tasti: con C (course) potrete cambiare rotta al sottomarino da 000 a 359 gradi, con D (depth) variare la profondità da 60 a 500 piedi, con S (speed) la velocità da 0 a 8 nodi, con P (periscope) alzare o abbassare il periscopio (solo se vi trovate a 60 piedi di profondità), con T (target) inquadrare automaticamente il bersaglio grosso e con E (escort) la scorta; infine, con 1 2 o 3 lanciare il numero scelto di siluri sull'obbiettivo.
Che cosa dovete fare per vincere? Come prima manovra sistemare la rotta in modo da farla coincidere con quella della nave da affondare, quindi portarsi il più vicino possibile ad essa e, al momento giusto, sancire i vostri ordini distruttivi. Se tutto va bene la vedrete esplodere e inabissarsi. A questo punto filate via rapidamente, se non volete incappare nelle cariche di profondità lanciate dalla scorta.
Qualora le nostre spiegazioni non fossero abbastanza esaurienti, all'interno della cassetta c'è tutto ciò che bisogna sapere per riuscire a portare a termine la missione, scritto, per giunta, in italiano. Il gioco è ben fatto sotto tutti i punti di vista, grafica compresa.

MARCO SQUINTANI

LE MEMORIE DI LAVORO

ROM

Dopo aver affrontato nel numero scorso l'argomento delle memorie volatili RAM passiamo stavolta a trattare quelle memorie che sono alla base di ogni sistema operativo e linguaggio di programmazione residenti all'interno del computer.
Che cosa vuol dire esattamente ROM? È presto detto. Dalle parti di Reagan e della Tatcher si traduce con Read Only Memory, ma, per chi non conoscesse l'inglese, ecco di seguito la traduzione italiana: memoria di sola lettura.
Anche a questo punto, tuttavia, le cose non sono ancora completamente chiare. Nell'intento di far luce sulla faccenda, possiamo aggiungere che le ROM contengono informazioni che possono essere solo lette e utilizzate, ma non modificate come nel caso delle RAM. Al contrario di queste ultime hanno poi la peculiare caratteristica di poter mantenere indeterminatamente il loro contenuto. Come tutte le memorie, anche le ROM sono misurabili quantitativamente in bytes.
È possibile con le ROM aumentare o modificare l'essenza stessa del calcolatore aggiungendo, per esempio, nuovi comandi (v. Extended Basic), utilizzare diversi linguaggi di programmazione (assembler, logo, Pascal, ecc.) o, addirittura, avere immediatamente disponibili interi programmi (v. Solid State Software).
Quando inseriamo una di queste cassette nell'apposita slot, sul video appare solitamente un piccolo menù con le due opzioni: 1 TI BASIC, 2 TI INVADERS (o altre simili). Perché questo? Semplicemente perché il nostro TI 99 vuol sapere che pesci prendere, o meglio, se deve andare a sinistra utilizzando le ROM del linguaggio residente, o andare a destra a combattere per voi contro gli alieni. Una piccola precisione: quel contenuto di memoria che ci fa «colloquiare» con il computer in modo comprensibile per entrambi si chiama — guarda caso — interprete. Tutte queste memorie, comunque, sono formate da una semplice matrice di diodi e la loro programmazione viene effettuata, senza possibilità di modifiche, facendo fondere, con l'applicazione di un campo di alta tensione, i diodi indesiderati. Le informazioni date in codice binario si ottengono analizzando la mancanza o la presenza dei collegamenti tra le linee.

LE PROM E LE EPROM

Le PROM (Programmable Read Only Memory) non sono altro che delle ROM vuote sulle quali è possibile immettere dati e programmi in permanenza tramite le sopraccitate procedure. Per poter modificare queste memorie, sono stati introdotti dei componenti che potevano essere svuotati dal loro contenuto, esponendo la matrice di diodi a un bagno di luce ultravioletta, attraverso un foro praticato alla sommità del dispositivo. Le memorie di questo tipo vengono indicate con la sigla EPROM, dove la E sta per Erasable (che significa cacellabile). Cosa non si farebbe per risparmiare...

MARCO SQUINTANI

COME COSTRUIRE UN ARCADE GAME:

I COMANDI

MAURO MATTIAZZI
M.M.G.

Eccoci ancora qui, ormai al terzultimo giro di boa.

A questo punto, potremmo già tirare qualche somma.
Dobbiamo dire a questo proposito che per noi della M.M.G. questa serie di articoli è stato un modo come un'altro per trovarci insieme a voi, e scoprire come il nostro «vecchio» TI99 sia ancora recente e all'avanguardia.

Tuttavia, nonostante siamo quasi alla fine del lavoro, abbiamo ancora davanti a noi delle tappe importanti da compiere assieme. Una di queste tappe è appunto quella dei comandi, infatti, abbiamo imparato a muovere sfondi e nemici, ma non siamo ancora in grado di muovere la nostra nave, o il nostro personaggio, e di sparare.

Iniziamo parlando del comando che fa girare tutto il sistema comandi: l'istruzione «CALL KEY».

Bisogna ammettere che anche qui il nostro 99 supera di gran lunga tutti gli altri, poiché alla «primitiva» efficienza di un «GET» o di un «INKEY» noi con una sola istruzione riusciamo ad analizzare lo stato di tastiera, ogni volta che l'istruzione viene richiamata.
Il formato più corretto è il seguente:

CALL KEY (Unità di tastiera, cod. CHR, STATO)
l'unità di tastiera: questo numero, compreso tra 0 e 4 rappresenta una delle 5 assegnazioni che si possono fare ai tasti, cioè a seconda del numero selezionato ci è possibile associare ai tasti della consolle, dei valori diversi, come mostrato nel manuale d'uso. Per un nostro utilizzo è utile sapere che la Tastiera «O» restituisce i codici ASCII standard.

Tastiera «1» e «2», rappresentano una suddivisione simmetrica della tastiera, cioè se volete fare un gioco a 2 e non avete i joystick, si selezionano queste 2 tastiere alternativamente, muovendo 2 caratteri.
È da notare che i tasti che sono simmetricamente allo stesso posto retituiscono lo stesso valore.
Tastiera «3» restituisce solo la maiuscola.
Vediamo ora qualche esempio:

```
TASTIERA «0»
  10 CALL CLEAR
  15 GOSUB 110
  20 RIGA = 12
  30 COLONNA = 16
 ·40 CALL HCHAR (RIGA, CO-
LONNA, 32)
  50 RIGA = RIGA + (K = SU)
- (K = GIU) - (RIGA < 2) + (RI-
GA > 22)
  60 COLONNA = COLONNA
+ (K = SIN) - (K = DES) - (CO-
LONNA < 2) + (COLONNA >
30)
  70 CALL HCHAR (RIGA, CO-
LONNA, 88)
  80 CALL KEY (O, K, S)
  90 IF S =O THEN 80
  100 IF (K = SU) + (K = SIN)
+ (K = DES) THEN 40 ELSE 80
  110 PRINT «SCEGLI I TASTI»
  120 INPUT «PER SALIRE»: SU
  130 INPUT «PER SCENDE-
RE»: GIU'
  140 INPUT «PER ANDARE A
DESTRA»: DES
  150 INPUT «PER ANDARE A
SINISTRA»: SIN
  160 RETURN
```

Spieghiamo il listato n° 1.
Alla riga 15, si richiama la subroutine che vi permette di selezionare i tasti che più vi piacciono per muovere il carattere sullo schermo, poi si stabiliscono le coordinate di

partenza e si cancella il vecchio carattere (riga 40) prima di calcolare le nuove coordinate alle righe 50 e 60.
Quindi alla riga 70 stampare il nuovo carattere, per il nostro 99 è un gioco da... calcolatrice!
Alla fine troviamo alla riga 80 la nostra CALL KEY.
Da notare che «S» variabile di stato, se è uguale a «0» nessun tasto è stato premuto, mentre viceversa procede a verificare che i tasti siano stati premuti. Se non premete i tasti selezionati, nulla accade.
Infine, un poco di attenzione al calcolo delle righe 50 e 60, poiché sappiamo che ad ogni condizione che è verificata il TI restituisce valore «-1» mentre restituisce valore «0» se non è verificata, con una semplice operazione algebrica di segni, a seconda del tasto premuto viene sommato o sottratto «1».
Allo stesso modo per il controllo del campo, cioè per impedire al carattere di uscire dal video, si utilizza lo stesso sistema cioè di sottrarre se il numero eccede e aggiungere nel caso il numero sia minore di 2.

```
TASTIERA «1» e «2»
  10 CALL LEAR
  20 DIM RIGA(2), COLONNA(2), NOME$(2)
  30 DATA 10,5,10,28
  40 FOR C=1 TO 2
  50 READ A,B
  60 RIGA(C) = A
  70 COLONNA(C) = B
  75 INPUT «NOME»: NOME$(C)
  80 NEXT C
  90 FOR C=1 TO 2
  95 GOSUB 255
 100 CALL KEY (C, K, S)
 110 IF K=18 THEN 160
 120 IF (K <> 5) + (K <> 3) THEN 300
 125 CALL HCHAR (RIGA(C), COLONNA(C), 32)
 130 RIGA(C) = RIGA(C) + (K = 5) - (K = 3) - (RIGA(C) < 3) + (RIGA(C) 20)
 140 CALL HCHAR (RIGA(C), COLONNA(C), 88)
 150 GOTO 300
 160 IF C=2 THEN 195
 170 CALL HCHAR (RIGA(C), COLONNA(C) + 1, 62, 22):
 180 CALL HCHAR (RIGA(C), COLONNA(C) + 1, 32, 22)
 190 GOTO 230
 195 FOR Z=COLONNA(C) -1 TO6 STEP -1
 200 CALL HCHAR (RIGA(C), Z, 60)
 210 NEXT Z
 220 CALL HCHAR (RIGA(C), 6, 32, 22)
 230 IF RIGA(1) RIGA(2) THEN 250
 240 CALL SOUND (1000, -7, 0)
 255 GOTO 300
 250 CALL HCHAR (1, 1, 32, 32)
 260 FOR Z=1 TO LEN (NOME$(C) )
 270 CALL HCHAR (1, 2+Z, ASC (SEG$ (NOME$, 2, 1) ) )
 280 NEXT Z
 290 RETURN
 300 NEXT C
 310 GOTO 90
```

Anche qui un po' di spiegazione:
il sistema utilizzato è quello precedente, in questo casi i caratteri da muovere sono due che vanno in alto e in basso. Al posto del movimento longitudinale abbiamo messo uno sparo.
Per il giocatore di sinistra i tasti sono i seguenti:

Q= SPARO
E= PER SALIRE
D= PER SCENDERE

per il giocatore di destra:

Y= SPARO
I= PER SALIRE
K= PER SCENDERE

Come avrete notato in questo caso la «S» viene ignorata, poiché se uno dei giocatori non preme di sicuro, o quasi, l'altro può farlo.

In questo modo, il programma legge una volta la tastiera «1» e una volta la tasiera «2», alternativamente, dando l'impressione di un moto continuo.
Inoltre potete notare che la CALL KEY è unica, poiché i codici dei tasti scelti, essendo simmetrici sono uguali.
Per darvi l'idea dell'alternarsi del ciclo, inserendo dei nomi all'inizio, od a ogni turno, verrà stampato il nome del giocatore. Questo naturalmente rallenterà il movimento.
Infine un cenno al comando dei joystick.
I joystick funzionano grosso modo come la CALL KEY, solamente che i valori sono fissi come potete leggere a pag. 94 del Manuale d'Uso. Lo sparo inoltre viene letto tramite la CALL KEY che specifica come unità di tastiera, o 1 o 2 a seconda del joystick usato, mentre il valore che viene restituito in K è uguale a 18. Ora come esempio possiamo sostituire il joystick alla tastiera, negli esempi precedenti, nel seguente modo:

ES. 1

```
  50 RIGA = RIGA + (K = 4) — (K = -4) - (RIGA < 2) + (RIGA > 22)
  60 COLONNA = COLONNA + (X = -4) - (X = 4) - (COLONNA < 2) + (COLONNA > 30)
  80 CALL JOYST (1, X, K)
  90 IF (X = 0) * (K = 0) THEN 80
```

ES. 2

```
 120 CALL JOYST (C, X, K):
 121 IF (X = 0) * (K = 0) THEN 300
 130 RIGA(C) = RIGA(C) + (K = 4) — (K = -4) - (RIGA(C) < 2) + (RIGA(C) > 22)
```

E con questo anche questa volta chiudiamo il discorso.

ALTA RISOLUZIONE PER TI 99 PIU' MINI MEMORY

Finora si é parlato dei vari modi grafici disponibili sul nostro TI 99; ora vi presentiamo una serie di routine in l/m per poter utilizzare da *basic* il BIT-MAP mode. Cioé diventa possibile utilizzare tutte quelle istruzioni di LINE, PLOT, CIRCLE disponibili in altri basic. La risoluzione dello schermo è, come tutti sapete, di 256x192 pixel, però per uniformarci allo standard di altri basic abbiamo preferito numerare i pixel da 0 a 255 e da 0 a 191, ponendo l'origine degli assi in basso a sinistra, lo schermo diventa così il 1° quadrante di un sistema di assi cartesiani. Per quanto riguarda il colore vi é la solita limitazione hardware cioè non vi sono limiti in verticale, mentre in orizzontale non vi possono essere piu di due colori, uno di fondo e uno di primo piano, per ogni sequenza di 8 pixel; in definitiva la risoluzione cromatica é di 32x192. Per ottenere queste caratteristiche di risoluzione la macchina usa quasi tutta la VDP ram per contenere le tavole dello schermo e del colore, lasciando ai programmi basic solo 2 kb che non sono molti ma permettono già di scrivere qualche piccolo programma dimostativo. L'unico modo per poter disporre di più memoria è quello di usare l'EXTENDED BASIC + 32 KB, in questo modo a patto di non usare molte stringhe, si possono scrivere programmi anche molto lunghi.

Passiamo ora a descrivere l'uso delle varie routine:

— CALL LINK («HGR»,col) serve a passare alla grafica in bit-map, pulisce la pagina grafica e setta il colore dello sfondo del colore indicato; può essere richiamata più volte all'interno di un programma al solo scopo di pulire la pagina grafica.

— CALL LINK («PLOTO», row, Col, clr) serve ad accendere il pixel alle coordinate row di riga e col di colonna e a colorarlo del colore clr

— CALL LINK («DRAW», incrY, incrX, clr, ...,) traccia un segmento che parte dall'ultimo punto che è stato plottato e arriva ad un punto che ha quelle stesse coordinate ma incrementate di quanto specificato nell'istruzione; segue poi il colore del segmento; i tre parametri, incremento Y (verticale), incremento X (orizzontale) e colore possono ripetersi per un numero massimo di 5 volte.

Ad esempio:

CALL LINK («PLOT», 70, 80,9)

CALL LINK («DRAW»,30, -20, 6, 0, 8, 7).

Queste due istruzioni tracciano due segmenti dal punto 70,80 a 100,60 e da 100,60 a 100,68, rispettivamente di colore blu e rosso.

— CALL LINK («LINE», Y1, X1, Y2, X2, clr) traccia una linea tra i punti di coordinate Y1,X1 e Y2,X2 colorandola del colore clr

— CALL LINK («CIRCLE», Y, X, RY, RX, clr) traccia un'ellisse di centro y,x e con un raggio verticale pari a RY e un raggio orizzontale pari a

GIOCHIAMO CON IL TI 99/4A

Un libro che non deve assolutamente mancare nella vostra libreria.

Potrete riceverlo a casa vostra a sole **L. 9500** + Spese di spedizione.

Per ogni richiesta di contrassegno, utilizzate il tagliando qui sotto spedendolo a:

NEWSOFT snc
via S. Jacini, 4
20121 MILANO

Vogliate spedirmi in contrassegno di L. 9500 + spese il libro: «GIOCHIAMO CON IL TI 99/4A»

Cognome .....

Nome .....

Indirizzo .....

.....

Cap. ..... Loc. .....

RX, se i due raggi sono uguali sarà tracciata una circonferenza anche se sullo schermo non sembrerà tale. Perché sembrino circonferenze il rapporto dei raggi dell'ellisse deve essere di 1.4 con quello verticale più grande.
— CALL LINK («POINT», Y, X, var) se il pixel alle coordinate Y,X è acceso alla variabile viene assegnato -1 altrimenti ∅.

Vi è ancora da aggiungere che prima di richiamare la routine HGR per passare in alta risoluzione bisogna fare due CALL LOAD e precisamente:

CALL LOAD (-32658,56,0,"",-32732,55,248) che servono a limitare l'area ram disponibile al basic. Un'altra osservazione riguarda tutte le routine; se si specifica il colore N.1 invece di una linea in colore trasparente le routine cancelleranno quanto era stato scritto prima riportando OFF quei pixel che erano accesi. È possibile così fare delle linee e poi, specificando il colore 1, cancellarle, semplicemente riscrivendovi sopra. Per tornare in modo grafico normale alla fine del programma basterà inserire prima dell'end questa istruzione, oppure digitarla con calma sulla tastiera una volta che il programma si sia fermato:

CALL PEEKV (-2048,A)

Anche se il programma si ferma per un errore o perché avete premuto CLEAR non fatevi prendere dal panico se lo schermo mostra cose strane, ma digitate con calma quanto riportato sopra, premete enter e tutto tornerà normale o quasi. Può darsi ad esempio che il cursore e le minuscole abbiano degli strani pattern, comunque tutto funzionerà normalmente.
Sulla cassetta dopo il programma da caricare in EASY BUG vi sono alcuni piccoli demo che utilizzano queste nuove istruzioni e che potete usare come esempio per capire il funzionamento delle singole istruzioni.
Il byte 29136 contiene il colore di sfondo con il quale vengono tracciate le linee. Se occorre un colore di sfondo diverso da quello trasparente basta fare un CALL LOAD (29136, clr) dove il colore è quello del basic meno uno, cioè se vogliamo il colore nero il cui codice è 2 dovremo inserire nel call load il numero 1. Per cancellare deve essere necessariamente messo a ∅.

GIAMPAOLO BOTTIN

«HELPER» TRADUZIONI IN CODICE MACCHINA

Il programma di nome HELPER, che trovate nella cassetta, vi fornirà un notevole aiuto per la traduzione dei codici, binari, decimali e esadecimali.
Esso può operare in tutti e tre i campi, traducendo perfino il numero inserito in nibble.
Il programma è di semplice utilizzo, esso si ramifica in tre parti distinte, per ogni ramo di traduzione si prevede la visualizzazione di tutti i campi con le rispettive traduzioni.
I campi di traduzione che possiamo scegliere sono i seguenti:

1) Hex. (esadecimale)
2) Dec. (decimale)
3) Bin. (binario)

Per selezionare il campo desiderato, basta premere il tasto corrispondente alla prima lettera della traduzione voluta.
Esempio: se vogliamo operare traduzioni esadecimali basterà premere la lettera H, quindi ci verrà richiesto il numero da tradurre.
La stessa regola vale naturalmente per gli altri campi di traduzione. Il programma è assolutamente protetto contro ogni inserimento di eventuali dati al di fuori di quelli consentiti per ogni campo.
Effettuata l'operazione di inserimento, il programma scriverà sullo schermo i risultati dell'elaborato.
Per primo troviamo il numero decimale e di fianco, possiamo ottenere la traduzione in nibble decimale dello stesso.
Alla riga successiva troviamo la traduzione esadecimale e di seguito quella binaria.
Terminata l'operazione, il computer ci riproporrà la fase di scelta traduzione lasciando la precedente traduzione sullo schermo per permetterci di poterla comodamente ricopiare.
Questo programma è composto da intricati algoritmi che lo rendono molto interessante anche dal punto di vista della programmazione.
Tenetelo sempre a portata di mano perché ne faremo largo uso nei prossimi appuntamenti con la rubrica EXTENDED + 32 Kbytes.

EZIO MONTINI

ROBOTIME

Questo mese, il gioco è in Extended. Tralasciando il titolo «vagamente» straniero, vi offriamo un gioco completamente nuovo. Come al solito il gioco è stato costruito sugli algoritmi illustrati nel precedente articolo, quindi tralasciamo la spiegazione del funzionamento già visto in precedenza e veniamo al gioco.

Nell'ideare questo Game, abbiamo dovuto optare per una scelta, poiché per gestire della grafica decente, era necessario rinunciare ad un po' di velocità di gioco. Noi abbiamo trovato una via di mezzo. Questo non significa che il gioco sia facile, significa solamente che ovviamente non ci saranno figure che sfrecciano o coincidenze immediate, tuttavia il programma soddisfa sia i fautori dei giochi veloci, sia gli amanti della grafica ultra fedele. Infine come chicca finale anche l'animazione delle figure, è in questo programma sufficientemente accettabile.

Il tema del gioco è piuttosto complesso: l'astronave deve addentrarsi in una feritoia spazio-temporale, dove ci è possibile raccogliere dei lingotti di «ULNOMANITE», materiale prezioso per la sopravvivenza umana.

Purtroppo questo minerale non esiste in natura, ma bisogna generarlo uccidendo i ROBOT che vivono all'interno della feritoia. Ogni robot colpito dal vostro proiettile viene trasformato istantaneamente in lingotto.

La feritoria è strutturata in stanze, ogni stanza è abitata da 5 robot sempre più veloci e cattivi. Non si può uscire dalla stanza se non si uccidono tutti i robot, che sono tutt'altro che indifesi. Una volta uccisi tutti i robot, bisogna passare e fermarsi qualche istante sopra i lingotti, affinché l'equipaggio li possa caricare a bordo; ogni lingotto caricato sono 10 punti. Anche qui il lavoro e tutt'altro che tranquillo, poiché come vengono uccisi i robot, compare una astronave, che cerca di impedirvi di recuperare i lingotti, lanciandovi contro dei MUTANTI CIBERNETICI, che mentre siete in movimento, non possono farvi niente, ma quando siete fermi, se vi toccano vi fanno esplodere facendovi perdere tutti i lingotti del quadro.

Se riuscite a raccogliere tutti i lingotti, vi viene assegnata una bandierina che vedrete nell'angolo in basso a sinistra. Tutte queste cose le dovete fare nel tempo che vi viene assegnato, a secondo del livello di difficoltà che sceglierete all'inizio. Ricordate che se il tempo finisce, non perdete una sola astronave, le perdete tutte, viceversa, una volta finito il quadro vi verrà dato altro tempo, che verrà sommato al residuo precedente.

Ogni 4 bandierine vincete una astronave, mentre ogni 3 quadri c'è un quadro Bonus. Nel quadro Bonus avete a disposizione trenta secondi per accumulare più punti possibili: si tratta in pratica di toccare con la vostra astronave, il Rotante Energetico, che vi ruoterà attorno per tutti i 30 secondi il maggior numero di volte possibile. Ogni contatto con il rotante vi frutterà 200 punti. In questo quadro come nella fase di recupero, il vostro cannone è disattivo.

Infine due parole sul come caricare il gioco e sui movimenti. Il gioco si compone di 2 programmi; caricato il primo, avremo la presentazione, e la definizione dei caratteri grafici.

Premuto un tasto, il programma vi chiederà di caricare il secondo, contenente il gioco vero e proprio.

Una volta caricato il gioco, è possibile scegliere di giocare o con il joystick o con la tastiera scegliendo i tasti che dovranno agire sui movimenti. Infine potrete scegliere su quattro livelli di difficoltà, i quali influiranno sul tempo a vostra disposizione.

Buon divertimento... ROBOTBUSTERS.

P.S. Prima di caricare il gioco, se l'avete, scollegate l'espansione.

MAURO MATTIAZZI
M.M.G.

UN'AVVENTURA NEL CALCOLATORE

Le avventure, di solito, siete abituati a viverle insieme al vostro TI 99 che è, di volta in volta, compagno o temibile avversario in tanti giochi.
Ma provate, questa volta, un gioco diverso, provate a scoprire la sua logica e assaporate il piacere della scoperta di un mondo affascinante ed estraneo.
È sera, state battendo distrattamente la tastiera del vostro calcolatore e, improvvisamente, vedete un messaggio sullo schermo del vostro televisore:

CPU ERRORE

Svanisce ogni immagine, lo schermo resta bianco anche provando a spegnere e riaccendere il calcolatore; capite che il guaio è grosso, battete a cos dei tasti e chiudendo gli occhi pensate in quale modo si possa riparare la consolle.
Voila! Una forza vi spinge all'interno del computer; riaprite gli occhi e vi trovate in una sala intensamente illuminata; questa sala rappresenta la tastiera, la principale via di comunicazione tre il mondo esterno e il computer. Da qui comincia il vostro viaggio nel mondo elettronico.
Ci sono tre porte sbarrate; una lettera, sulla porta centrale, indica la direzione in cui state guardando.
Ogni porta è bloccata da una certa combinazione di circuiti formati da porte logiche AND, descritte da un simbolo con l'angolo squadrato, e da porte OR a forma di freccia.
La porta AND richiede, per funzionare, che entrambi i suoi ingressi siano attivati, la porta OR che lo sia soltanto uno di essi.
Premendo i numeri adatti posti alla sinistra dello schermo si arriverà al termine del circuito logico e il blocco della porta verrà tolto. A questo punto, battendo i tasti E, W, S, N, corrispondenti ai vari punti cardinali, vi sarà possibile seguire una certa direzione alla ricerca della CPU guasta.
Troverete molte porte e molti circuiti da sbloccare, attenzione; il computer tiene conto del numero di tentativi, se essi saranno troppo numerosi perderete la possibilità di uscire da questo mondo elettronico.
Attenti poi a settori guasti del disco, non fidatevi troppo e... in bocca al lupo.

PAOLO CIVARDI

Descrizione del programma.

100 - 170	Inizializzazione variabili.
180 - 190	Direzione.
200 - 230	Ingresso in un settore difettoso del disco.
240 - 270	Porta aperta.
280 - 340	Visualizzazione delle sale del computer.
350 - 410	Visualizzazione porte logiche e attesa di input.
420 - 690	Logica di controllo della porta.
700 - 830	Ritrovamento della CPU. Fine gioco.
840	Scansione della tastiera.
850 - 1230	Serie di Data per la grafica.

Avete acquistato il Numero di Marzo?

Ma sapete che sono usciti i Numeri di Giugno Settembre Ottobre Novembre Dicembre Gennaio Febbraio e Marzo? Per riceverli direttamente a casa vostra, scriveteci! Ve li invieremo in contrassegno a L. 15000 (+spese cadauno).

TI 99 NEWSOFT
Via S. Jacini, 4
20121 MILANO

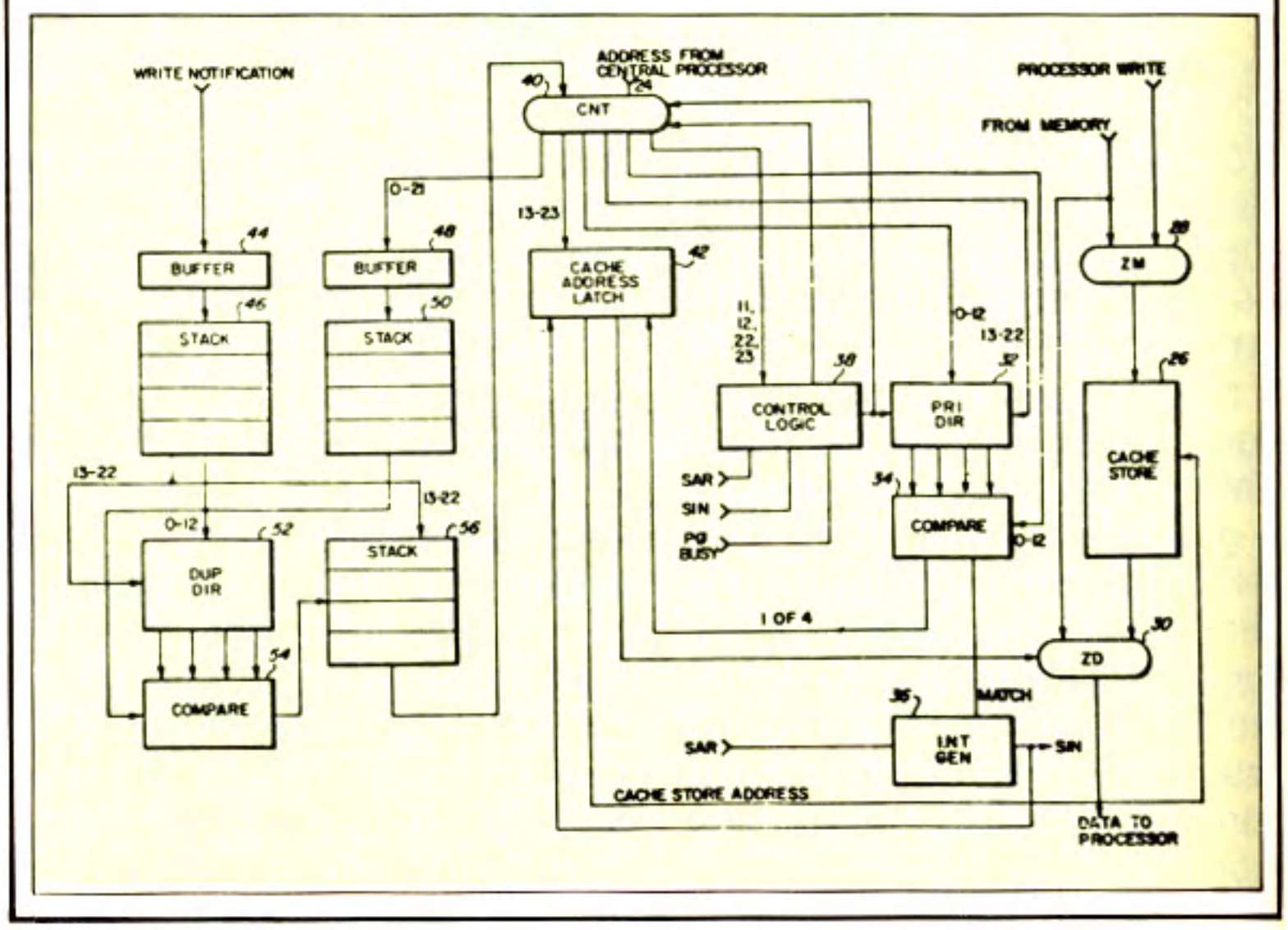
Computer con memoria nascosta

Brevetto Honeywell Inf. Systems Inc. 12037466.

Un processore presenta una memoria nascosta 26, di 512 posizioni di memoria ognuna, includente 4 livelli, ognuno dei quali memorizza al blocco di dati.

Ogni livello di ciascuna posizione ha un rispettivo flip-flop (multivibratore bistabile) pieno/vuoto associato con esso nella logica di controllo 38. Un indice principale ha 512 colonne, ognuna di 4 livelli per memorizzare gli indirizzi di sezione dei blocchi nella memoria nascosta 26 — ogni sezione di memoria principale contiene 512 blocchi. Un indirizzo assoluto di «indirizzo di sezione» + «indirizzo di blocco» entro la sezione, ha il suo indirizzo di blocco che viene usato per la ricerca dei 4 indirizzi di sezione nella corrispondente colonna dell'indice; un confronto con l'indirizzo assoluto di sezione indica che il blocco richiesto è in memoria nascosta e identifica il suo livello.

Un doppio indice 52 duplica i contenuti dell'indice principale 32. Le scritture provenienti da altre sorgenti hanno i loro indirizzi controllati rispetto al doppio indice. Se c'è un confronto, il corrispondente multivibratore bistabile pieno/vuoto associato con l'indice principale e con la memoria nascosta viene azzerato, in modo che il blocco che è stato modificato dalla scrittura viene effettivamente rimosso dal sistema a memoria nascosta.



E LE STELLE

STANNO

A PARLARE

Ognuno di noi prova un certo fascino, o almeno curiosità, per una scienza così antica ma sempre attuale come l'astrologia.
Anche il TI 99 non è rimasto indifferente agli astri e alla loro influenza sui nostri destini ed ecco allora un programma che vi permetterà, in base alla data e ora di nascita, di stabilire il vostro ascendente, il colore, la pietra, il numero ed altre caratteristiche specifiche del segno zodiacale sotto il quale siete nati.
Potrete così «giocare» a fare gli astrologhi per voi, la vostra famgilia e tutti gli amici che lo desideranno, lasciando che una volta tanto non sia solo il computer e l'elettronica a parlare, ma anche le stelle.

DARIO CIMBERLE

DIAGRAMMI A SETTORI CIRCOLARI

Assieme ai grafici a strisce, una delle rappresentazioni grafiche più comuni è certamente quella per settori circolari. In libri, giornali e riviste non è raro veder rappresentare in questo modo, ad esempio, i risultati elettorali o le quote di mercato conquistate da alcune società in un determinato settore produttivo. Il loro uso è particolarmente indicato quando si vuole mettere in evidenza il valore percentuale di certe quantità. Infatti, l'intera superficie del cerchio corrisponde al 100%, mentre ogni settore deve avere un'ampiezza proporzionale al valore percentuale del dato che rappresenta. Da un punto di vista matematico le difficoltà sembrano irrilevanti; infatti all'1% corrisponde un angolo di 3.6 gradi e con una semplice moltiplicazione si può calcolare l'ampiezza di un settore corrispondente ad una qualsiasi altra percentuale. Tuttavia la realizzazione al computer richiede conoscenze ben più profonde e necessita dell'uso di funzioni trigometriche ed analitiche, tanto che il programma non risulti del tutto banale.
Innanzi tutto sarebbe gradito l'uso dei colori, in modo che ogni settore venga contraddistinto con uno dei 16 disponibili. Purtroppo la cosa non è fattibile poiché ogni carattere può assumere soltanto due colori, uno per il foreground ed uno per il backgroud. Se nella piccola superficie corrispondente ad un carattere convergono tre settori oppure due settori e lo sfondo, la rappresentazione diventa impossibile. Questo programma utilizza pertanto due soli colori, tuttavia usa la grafica in alta risoluzione. In TI BASIC puro e semplice non sarebbe possibile utilizzare tutto lo schermo in alta risoluzione; tuttavia, sfruttando i caratteri programmabili, si può ottenere tale modo grafico su una piccola porzione dello schermo, ampia 8×8 caratteri o, se vogliamo, 64×64 pixel. Una finestra più ampia per l'alta risoluzione

Principali routine utilizzate nel programma

230- 360	caricamento di una tabella per effettuare l'operazione logica OR tra due semibyte espressi con caratteri esadecimali. L'operazione consente di mettere in ON un pixel senza alterare gli altri appartenenti allo stesso carattere.
370- 870	definizione dei caratteri grafici. Il pattern dei caratteri è memorizzato nel vettore CH$().
880- 1040	titolo.
1050- 1160	ridefinizione di alcuni caratteri per annullare i settori presenti nel titolo.
1170- 1530	input, modifica e conferma dei dati.
1540- 1640	stampa dei valori percentuali.
1650- 2360	routine per la stampa del diagramma a settori circolari. Le linee 2030-2300 sono organizzate in subroutine e servono a pozze in ON un unico pixel nel diagramma.
2370- 2520	menu per la scelta delle possibili continuazioni del programma.
2530- 2640	routine per la tabulazione dei dati.
2650- 2720	routine per ripresentare il grafico sullo schermo.
2730- 2830	routine per il calcolo delle misure angolari dei settori.
2840- 3010	subroutine per tracciare il cerchio in cui verranno segnati i settori.
3020- 3070	subroutine per simulare l'istruzione PRINT AT.
3080	fine del programma.

Principali variabili usate nel programma

S$(,)	tabella o matrice bidimensionale per eseguire l'operazione logica OR tra due caratteri esadecimali.
CH$()	vettore per memorizzare la forma dei caratteri che costituiscono la finestra ad alta risoluzione.
J,K	variabili usate nei cicli FOR NEXT. K è usata anche nell'istruzione CALL KEY per memorizzare il numero di codice ASCII del tasto premuto.
RIGA, COLO	numero di riga e di colonna per le subroutine.
N	numero dei dati da inserire nel programma.
D$(),Q()	vettori per memorizzare le descrizioni e le quantità corrispondenti.
SS	somma totale delle quantità. È necessaria per il calcolo dei valori percentuali.
P()	valori percentuali corrispondenti alle varie quantità.
PI	valore approssimato di pi greco.
RAD	misura, in radianti, dell'angolo al centro dei vari settori.
W	variabile di lavoro per salvare il valore del coseno dell'angolo.
X1,Y1	coordinate del punto dal quale si inizia a tracciare un lato del settore.
X2	ascissa del punto d'arrivo.
X,Y	coordinate del punto che sta per essere attivato nel diagramma.
FLAG	variabile flag che indica se va colmato l'eventuale vuoto esistente tra due punti consecutivi.
RO,CO	numeri di riga e colonna relativi alla finestra ad alta risoluzione.
LN	indica quale degli 8 byte che formano un carattere contiene il bit corrispondente al pixel da attivare sul grafico.
PO	indica la posizione del semibyte, cioè a destra o a sinistra.
BI	indica quale bit deve essere attivato all'interno del semibyte.
B$	contiene il carattere esadecimale associato ad un semibyte.
ET	indica l'elemento di tabella da richiamare come prodotto dell'operazione OR tra due catteri esadecimali.
NC	numero di codice ASCII del carattere contenente il pixel da porre in ON.

avrebbe comportato la rinuncia dei caratteri per il testo.
L'alta risoluzione presuppone che si possa mettere in un pixel senza alterare gli appartenenti allo stesso carattere.

In Extended BASIC, per la presenza dell'istruzione CALL CHARPAT, la ruotine risulterebbe al quanto più veloce, tuttavia si sa che con il BASIC esteso non sono utilizzabili i caratteri con codice ASCII superiore al 143.
Ma veniamo all'uso del programma. Immaginiamo di voler rappresentare con un diagramma a settori circolari le vendite di personal computer in Italia, nell'anno 1983. Limitandoci alle tre maggiori società, riserveremo una quarta voce per rappresentare unitamente tutte le altre. Alla richiesta del numero dei dati risponderemo pertanto 4. Per ragioni di tabulazione non è possibile inserire più di 10 dati. Noi scriveremo ad esempio:

1) OLIVETTI, 17060
2) APPLE, 6300
3) IBM, 4900
4) ALTRI, 23000

I dati vengono poi tabulati ed il programma rimane in attesa di conferma. Se ci si accorge di un errore non sarà necessario riscrivere tutto daccapo ma sarà sufficiente riscrivere la sola linea incriminata.
Se tutto è OK vengono tabulati i dati con i valori percentuali accompagnati dal diagramma a settori circolari. Da qui si accede ad un menu dal quale si può tornare alla tabella iniziale o al grafico oppure passare ad una sezione in cui vengono fornite, in gradi decimali, le ampiezze di tutti i settori. Ciò può risultare utile qualora si desideri riprodurre il disegno con riga e compasso, ricordando tuttavia che tutti i valori sono approssimati per difetto a meno di 1/10.

SERGIO BORSANI

Minicalc

Questa volta, invece che parlare di statistica, pubblichiamo un programma, tratto da Home Computer, che con la statistica potrebbe, e deve, avere a che fare, ma che può comunque servire ad una infinità di applicazioni più generali.
Tutti noi abbiamo sentito parlare di quei tabelloni elettronici chiamati, in America, *spreadsheet*, su cui è possibile eseguire, in modo simultaneo, diversi calcoli matematici e che possono, si dice, usando un termine di alcuni anni fa, quando il fenomeno computer era ancora visto in modo un pochino pioneristico, trasformare un elaboratore in una macchina decisionale.
Questo programma, pur se scritto in Basic, e quindi con tutte le limitazioni di lentezza insite in tutti i Basic, anche quelli in dotazioni su macchine diverse dalla nostra, ci darà, finalmente, la possibilità di cominciare ad usare uno di questi famosi spreadsheers che hanno contribuito a creare la fama degli home computers.
Questo programa, è chiaro, non è *Visicalc*, non è *Multiplan* o *Lotus* ma, sono certo, vi aiuterà lo stesso nel curare i vostri conti di casa, le vostre previsioni statistiche e non so ancora cosa altro la fantasia o le propensioni personali potranno suggerire.
Con Minicalc è possibile assegnare ad ogni riga una certa formula che verrà impiegata per tutte le colonne della riga stessa.
Ad ogni riga può essere assegnato un nome; si può decidere se essa debba essere, o non essere stampata; nel caso in cui righe intermedie siano usate soltanto per l'esecuzione di calcoli intermedi e non debbano pertanto essere visualizzate nella stesura finale del rapporto. Si può, inoltre, fissare il numero totale delle colonne e stabilire una colonna dei totali.

Descrizione delle operazioni

All'inizio appare sullo schermo, sotto la dicitura "nome riga", un cursore bianco, uno sprite, che attende l'assegnazione del nome della riga; l'assegnazione del nome non è obbligatoria ma, nel caso avvenga, il nome potrà avere una lunghezza massima di nove caratteri, ogni parola più lunga verrà perciò troncata fino ad raggiungere questa dimensione.
A questo punto si può, mediante l'uso delle frecce, spostare il cursore e immettere i dati numerici desiderati; dati che, anch'essi, avranno una lunghezza massima di sette caratteri, compreso il punto.
Ogni volta che si lascia un determinata cella, o posizione, caratterizzata da un numero di riga e di colonna, i dati all'interno della cella vengono memorizzati e saranno sempre disponibili per tutto il tempo in cui il calcolatore resta acceso e per sempre, nel caso i dati vengano registrati su disco o cassetta.

Modo logico

Nel modo logico vengono definite le operazioni, o equazioni, necessarie allo svilup-

po dei risultati finali che appariranno sulla tabella. Durante questa fase è possibile assegnare il nome logico del modello, l'intestazione, denominare le righe, definire quale sarà la colonna dei totali e l'ultima colonna utile per elaborare i dati, eccetera.
Una volta che si è scelto di entrare nel modo logico, si vedrà scomparire la tabella e apparirà il segnale > sul fondo dello schermo.
Nel modo logico esistono due classi di comandi operativi. Essi sono i comandi immediati e quelli di funzione logica.

I comandi immediati sono:

NUOVO - con questo comando sono permesse quattro opzioni.

1) Dati - Scegliendo dati vengono eliminati dalla memoria tutti i dati numerici.

2) Logico - Con questo comando viene cancellato il modello logico impiegato.

3) Entrambi - Dati e modello logico sono cancellati.

4) Cancella - Il comando NUOVO non è più valido e si ritorna al modo logico senza rendere operanti le modifiche.

LIST - Con questa istruzione si ottiene sul video un listato del modello logico attuale.
Il primo elemento della lista è sempre il nome logico dato al modello. Qualora esso non sia stato assegnato appare la scirtta NOME LOGICO È senza intestazione.
Poi appare COLONNA TOTALE È n; se "n" è posto uguale a zero allora la colonna dei totali verrà disattivata. Il valore, per difetto, di "n" è sempre fissato inizialmente uguale a 13.
Segue poi la scritta ULTIMA COLONNA È "n" dove n indica l'ultima colonna in cui è possibile immettere dati. È, ovviamente, opportuno che questo valore sia inferiore a quello della colonna dei totali, il valore per difetto di questa colonna è 12.
Le altre voci listate dipendono dai vari comandi assegnati precedentemente, se per esempio, è stato assegnato un nome riga, quella riga verrà listata con il suo nome.
STAMPA - Agisce come LIST ma il listato è indirizzato alla stampante invece che allo schermo.
MUOVI x IN y - Permette lo spostamento di tutti i dati della colonna x nella colonna y. I dati della colonna x non saranno cancellati e saranno duplicati nella colonna y.
FINE - Con questa istruzione si esce dal modo logico e si torna al tabellone principale.
I comandi di funzione logica, quelli che non hanno cioè azione immediata, come quelli precedentemente descritti, e permettono di intervenire sul modello sono:
NOME LOGICO È - Questo comando permette di creare, o cambiare, il nome da dare al modello logico. Lo stesso nome verrà assegnato come nome file quando decideremo di registrare il tabellone elettronico su disco o cassetta.
COLONNA TOTALE È - Stabilisce la colonna dei totali; se è aggiunto OFF allora la colonna dei totali è disabilitata.
L'ULTIMA COLONNA È - È l'ultima colonna su cui si possono scrivere dei dati.
r È n - Serve a denominare ogni riga della tabella. "r" è il numero di riga e "n" è il nome da dare alla riga.
r = eq — "eq" è l'equazione assegnata alla riga determinata. L'equazione può derivare da altre righe della tabella, da numeri reali, o costanti, da un operatore, chiamato LAG, che permette di accedere ad altre colonne, da operatori aritmetici.
Le regole da rispettare per la definizione delle equazioni sono le seguenti:
1) — È possibile specificare una riga impiegando solo il numero identificatore della riga stessa. Per esempio 3 = 1 + 2 significa che alla riga 3 è assegnato il valore ottenuto sommando la riga 1 con la 2 e che l'equazione è valida per tutte le colonne delle righe considerate.
Gli spazi che avrete notato nella descrizione dell'equazione sono necessari e vanno usati per tutte le operazioni; è in questo modo che il calcolatore riconosce l'ordine logico delle operazioni da eseguire.
2) — Le costanti numeriche vanno poste tra parentesi. Se, per esempio, prendiamo l'istruzione 3 = 1 + 2 * (11.21) otterremo alla terza riga il valore della riga uno sommato a quello della due moltiplicato per la costante 11.21.
Importante notare che le righe 1 e 2 vengono sommate

tra di loro prima di essere moltiplicate per la costante in quanto non vengono rispettate le solite regole di priorità dei segni; le operazioni sono eseguite sempre a partire dalla prima a sinistra e proseguendo verso destra.
L'operatore logico LAG permette di impiegare dati da una colonna precedente quella attualmente in uso.
Quando LAG è posto davanti ad un numero di riga si ottiene il dato proveniente dalla stessa riga ma da una colonna precedente. Per esempio: 5 = 1 + LAG 2 significa che la riga 1, della colonna attualmente in uso, sarà aggiunta alla riga 2 della colonna immediatamente precedente quella attuale.
Se la colonna che si sta calcolando è la sesta, LAG 2 farà sì che venga impiegato il valore proveniente dalla quinta colonna della riga 2.
Un altro esempio: scrivendo 5 = 4 + LAG 5 otterremo nella colonna 5 il totale della riga 4.
Quattro sono gli operatori aritmetici che si possono impiegare nelle operazioni:

\+ Somma
— Sottrazione
* Moltiplicazione
/ Divisione

Se comunque analizzerete il listato, noterete come sia estremamente agevole aggiungere, in caso di necessità, altri segni come la radice quadrata, operazioni trigonometriche, ecc.
Una equazione può essere lunga un massimo di 127 caratteri, o quattro linee video (la lunghezza massima di un stringa permessa dal TI 99/4A). Nel caso vi serva una operazione complessa, è comunque possibile dividerla in più righe, ottenendo così lo stesso risultato.
Consigliamo, tuttavia, di controllare, mediante i comandi LIST, e eventualmente, STAMPA, tutte le operazioni logiche date al modello; sarà così più semplice verificare la corretta accettazione, da parte del computer, di esse.

Opzione di calcolo

Premendo il tasto corrispondente a questa opzione il calcolatore incomincia a calcolare il modello logico usando i dati contenuti nel tabellone. Tutte le operazioni vengono eseguite partendo dal numero di riga più basso.
Con questa opzione viene, automaticamente, assegnata una riga dei totali. Se però è stata disabilitata la colonna dei totali, la colonna 13, e non è stata predisposta una colonna alternativa, allora, nella riga dei totali, non verrà eseguito alcun calcolo.

Caricamento dei file di dati

Scegliendo questo comando avremo tre possibilità di caricamento dati.
1) Caricamento dei dati.
2) Caricamento del modello logico.
3) Caricamento di entrambi.
Il tabellone viene caricato integralmente ed è pronto a subire eventuali modifiche.
Dopo aver scelto l'opzione a noi più conveniente verrà richiesto il nome del file da caricare. Lo stesso nome dovrà, naturalmente, essere stato registrato in precedenza insieme allo *spreadsheet*.

Registrazione.

La presentazione è analoga a quella descritta prima per la registrazione.
L'unica differenza è costituita dal fatto che il programma usa come nome da assegnare al file il nome logico precedentemente assegnato. Se questo manca, in questa fase sarà richiesto, oppure, se si desidera cambiarlo, si può, in questo momento decidere il nome alternativo.

Cancellazione dei dati.

Per usare questa funzione, è sufficiente porre il cursore sopra il dato da cancellare e premere il tasto associato.
Se il campo è numerico esso verrà posto uguale a zero, se, invece, è letterale, al suo posto comparirà una stringa vuota.

Stampa.

L'opzione di stampa permette di avere una copia su carta del tabellone.
Prima che l'opzione di stampa abbia inizio, viene richiesto il titolo, la data, e il numero di righe da stampare. È importante notare che il programma propone il nome dell'interfaccia da usare: è tassativo rispondere allo stesso modo, battendo RS232.BA = 4800.DA = 8 (come, del resto, è necessario fare durante il caricamen-

SPESE CASA 1984 7.2.85

PAGINA 1

RIGA	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	GAS	.00	.00	34.10	.00	.00	29.20	.00	.00	.00	27.00	.00	.00	90.30
2	SIP	56.00	.00	40.00	.00	41.00	.00	36.00	.00	23.00	.00	35.00	9.00	240.00
3	LUCE	.00	.00	27.80	.00	27.85	.00	27.80	.00	30.25	.00	26.40	.00	140.10
4	AFFITTO	506.85	.00	.00	555.60	.00	555.60	.00	.00	555.60	.00	.00	.00	2173.65
5	RISCALD	340.00	.00	.00	170.00	.00	.00	.00	.00	400.00	.00	.00	.00	910.00
6	SPESE	127.65	.00	.00	175.00	.00	175.00	.00	.00	175.00	.00	.00	.00	652.65
7		.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
8		.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
9		.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
10		.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
11		.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
12		.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
13		1030.50	.00	101.90	900.60	68.85	759.80	63.80	.00	1183.85	27.00	61.40	9.00	4206.70

```
NOME LOGICO E'
COLONNA TOTALE E' 13
L'ULTIMA COLONNA E'12
1 E' A
2 E' B
12 E' TOTALI
13 = 1 + 2
```

I TASTI IN FUNZIONE

Premendo FCTN	Si ottiene la funzione
1	Caricamento dei dati da disco o cassetta.
2	Registrazione su disco o cassetta.
3	Cancellazione battuta o campo.
5	Calcolo del modello logico.
6	Stampa della tabella.
7	Si entra nel modo logico.
9	Si esce dal programma Minicalc.

FTCN	Funzione
E	muove il cursore in lato di una cella.
S	muove il cursore a sinistra di una cella.
D	muove il cursore a destra di una cella.
X o ENTER	muove il cursore in basso di una cella.

premendo CTRL

CTRL	Funzione
E	muove il cursore in alto di 5 celle
S	muove il cursore a sinistra di 3 celle
D	muove il cursore a destra di 3 celle.
X	muove il cursore in basso di 5 celle.

to o la registrazione dei dati battendo DSK1 e nome file o CS1).

Se viene messo un asterisco * prima o come unico carattere del nome riga, quella determinata riga non verrà stampata sul foglio. Ciò per permettere di usare alcune righe per i calcoli intermedi; calcoli che non necessariamente debbono apparire sul documento finale.

Modalità di stampa.

Per la stampa del tabellone, come si vede nell'esempio, è stata impiegata una EPSON MX 80.

Per poter stampare tutte le 13 colonne è stato necessario ricorrere alla stampa in modo condensato, questa forma grafica si ottiene sulla EPSON, e sulla maggior parte delle stampanti in commercio, usando CHRL$ (15); con questo comando si dispone di 132 colonne sufficienti a stampare in una sola pagina la tabella.

Se disponete di una stampante di tipo diverso da quella citata consultate il manuale d'uso per conoscere le modalità di stampa in modo condensato e cambiate, di conseguenza, il carattere della riga 1440.

Per coloro che dispongono dell'estensione di memoria.
Il numero massimo di righe utilizzabili con 16 K di base è di 2 mentre le colonne sono limitate a 13.

Usando l'espansione di memoria a 32 K del Texas si può agevolmente superare questo ostacolo.
È però necessario cambiare alla riga 190 il valore di A ponendolo uguale al numero totale di righe che si desidera avere.
Bisogna poi cambiare B fissandolo al numero di colonne necessarie. E modificare i valori di dimensionamento di D$ () e E$ ().
La matrrice J () dovrà aver fissato il primo valore al numero di righe necessario e il secondo numero posto uguale al totale delle colonne desiderate.
È tutto ma credo sia sufficiente per determinare il vostro interesse.

PAOLO CIVARDI

DICHIARAZIONE DEI REDDITI 1984

E' disponibile un programma
che finalmente vi guiderà alla determinazione
del reddito per i Mod. 101 e 740.

Il programma gira in Extended Basic
e costa Lit. 35.000 + spese di spedizione.

PER RICHIEDERLO
UTILIZZATE LA CARTOLINA
TI 99 MARKET

TI~99/4A

COMPU MARKET

TI~99/4A

- Consolle Home Computer TI-99/4A - 16 bit - 16 KB, completo di alimentatore e modulatore per collegamento TV con 2 moduli SSS gioco/educativi
  L. 255.000
  (abb. **L. 235.000**)

Solo per i nostri lettori, con condizioni speciali per gli abbonati a TI-99 NEWSOFT, il TI-99 Compu-Market vi offre anche questo mese i suoi prodotti super affidabili. Tutti i prezzi sono comprensivi di IVA, ma escludono i costi di spedizione e imballo. Il pagamento lo potrete effettuare contrassegno al ricevimento della merce. Per le ordinazioni usate la cartolina TI-99 MARKET allegata a questa rivista.

- Modulo **Extended Basic**
  L. 195.000
  (abb. **L. 175.000**)

- Espansione di memoria esterna (non richiede il BOX di 32 Kb RAM
  L. 220.000
  (abbonati **L. 200.000**)

- Cavetto adattatore Joysticks di qualsiasi tipo (Spectravideo...)
  L. 30.000
  (abb. **L. 28.000**)
- Joystick singolo
  L. 23.000
  (abb. **L. 21.000**)

Offerte speciali

- TI LOGO in italiano - Mondadori
  L. 195.000
  (abb. **L. 175.000**)
- TI LOGO in italiano Mondadori + espansione di memoria esterna da 32 Kb con libro: Impariamo a programmare in Basic
  L. 390.000
  (abb. **L. 365.000**)

- Registratore IRRADIO RE 730
  L. 72.000
  (abb. **L. 69.000**)
- Cavo di collegamento al registratore
  L. 12.500
  (abb. **L. 11.500**)

Interfaccia parallela esterna per il collegamento di stampanti

L. 210.000
(abbonati **L. 195.000**)

Cavo di collegamento interfaccia parallela esterna con qualsiasi stampante tipo Centronics mt. 1
L. 40.000
(abbonati **L. 37.000**)

TI-99/4A COMPU MARKET

DESCRIZIONE	PREZZO	PER GLI ABBONATI
Stampante Seikosha GP500A matrice 5x7, 50 crt/sec 80 colonne, normale ed espanso, grafica	615.000	560.000
Stampante EPSON RX80, 100 crt/sec grafica, 80/132 colonne bid. ott.	850.000	780.000
Stampante EPSON RX80 F/T come sopra gestisce anche foglio singolo	970.000	890.000
Monitor colori Hantarex CT900/2 con audio, 15" e cavo collegamento	640.000	590.000
Modulo TERMINAL EMULATOR	51.000	47.000
Modulo PRG gest. dati personali	51.000	47.000
Modulo ADD-SUB A/1 - educativo	20.000	18.000
Modulo MATH GAMES 6 - educativo	20.000	18.000
Modulo NUMERATION 1 - educativo	20.000	18.000
Modulo DEMOL. DIVISION - educativo	20.000	18.000
Modulo MATH GAMES II - educativo	20.000	18.000
Cassetta VIRGIN ROBOPODS	17.000	15.000
Cassetta VIRGIN FUN PAK	17.000	15.000
Cassetta VIRGIN FUN PAK 2	17.000	15.000
Cassetta VIRGIN FUN PAK 3	17.000	15.000
Cassetta BLUEGRASS corsa ippica	11.000	10.000
Cassetta DECATHLON	12.000	11.000
Cassetta GOBLINS REVENGE labirinto	11.000	10.000
Cassetta SNOW TREK corsa di slitte	11.000	10.000
Cassetta UP PERISCOPE batt navale	11.000	10.000
Cassetta FAMILY GAME 4 giochi	11.000	10.000
Cassetta TI 99 CALC	35.000	31.000
Cassetta determinazione del reddito	35.000	31.000

Nel programma STRIKE DEFENCE pubblicato nel numero di marzo, lato BASIC, della cassetta, per evitare la risposta MEMORY FULL al momento del "RUN" togliere le prime due righe:

```
1 SIZE = SIZE + 8
2 GOSUB 1
```

TI 99 NEWSOFT

MAGGIO 1985 L. 10.000

NUMERO NOVE

EXTENDED + 32 K: Come ottenere il Text mode

TI 99/4A: Un valido aiuto nella scelta della vostra autovettura

IN SOFTWARE & SOFTWARE:
Monster Manor - Album - Boumcer 3D

SOMMARIO

CERCHIAMO

per sviluppo software

PROGRAMMATORE

disponibile anche tempo pieno.

Rivolgersi in Redazione

riferimento software

□ **TI BASIC**
Come sfruttare i tasti di controllo della tastiera — pag. 2

□ **EXTENDED**
Animazione degli sprites — pag. 3

□ **TI 99/4A**
Esaminiamo insieme la "carrozzeria" del TI 99 — pag. 5

□ **EXTENDED - 32 K**
Vi spieghiamo come ottenere il text mode — pag. 7

□ **TEXT MODE**
Tre nuove istruzioni — pag. 10

□ **STATISTICA**
La regressione lineare — pag. 13

□ **PROGRAMMAZIONE**
Come costruire un arcade game: questa volta uno speciale dedicato ai labirinti — pag. 15

□ **3-D MONSTERS MANOR**
Il gioco del labirinto in TI BASIC — pag. 18

□ **RECENSIONI**
Parliamo della cassetta "Snow trek" — pag. 21

□ **BOUNCER**
Un gioco dell'altro... mondo — pag. 22

□ **ALBUM**
Un programma dedicato ai bambini in età prescolare — pag. 23

□ **AIUTO ALLE DECISIONI**
Il vostro TI 99 vi consiglia per l'acquisto di un'autovettura — pag. 24

□ **LA POSTA** — pag. 27

□ **LA VIGNETTA** — pag. 28

□ **PICCOLI ANNUNCI** — pag. 29

□ **COLLABORATE CON NOI** — pag. 31

□ **TI 99 MARKET** — pag. 33

TI 99 NEWSOFT
ANNO 2 - Numero nove
MAGGIO 1985
MENSILE
Direttore responsabile:
ANTONIO C. LOSITO
Coordinatore Editoriale:
VINICIO CIVININI
Direttore Amministrativo:
ENRICO CALLERIO
Art Director:
BEPPE RE-FRASCHINI
Illustrazioni:
MARIA TERESA COCOZZA
Segreteria:
CRISTINA FRISON
Redazione:
FIORENZA AURIEMMA
Collaboratori:
SERGIO BORSANI,
GIANPAOLO BOTTIN,
DANIELE CATALFAMO,
PAOLO CIVARDI,
STEFANO DELLI PONTI,
FRANCESCO MACALI,
MAURO MATTIAZZI,
EZIO MONTINI,
LEVIO PEROTTI,
RICCARDO ROSSI,
MARCO SQUINTANI,
ALBERTO STRAFILE,
SEBASTIANO TOMASELLO
TI 99 NEWSOFT è una pubblicazione della **NEWSOFT snc.**
via Stefano Jacini 4 - 20121 MI
Tel. 02/807464 - Telex 324284
REGISTRAZIONE AL TRIBUNALE DI MILANO N. 323 DEL 7.7.84

COPYRIGHT NEWSOFT snc
TUTTI I DIRITTI RISERVATI
Manoscritti e foto originali, anche se non pubblicati, non si restituiscono ed è vietata la riproduzione anche parziale di testi, fotografie e listati.

Pubblicità: AV STUDIO
Via Giuseppe Meda, 23
Tel. 02/8357155
Abbonamenti: l'abbonamento annuo a 11 numeri di *TI 99 NEWSOFT* costa L. 95.000 per l'Italia; L. 160.000 per l'Europa (spedizione aerea); L. 200.000 per tutti gli altri paesi.
L'importo, per assegno non trasferibile o per vaglia postale, va inviato a NEWSOFT snc, via S. Jacini 4 - 20121 Milano.
Stampa: CGB, Albegno di Treviolo (BG)
Concessionaria per la distribuzione:
EUROSTAMPA Srl - C.so Vittorio Emanuele 111 - 10123 Torino - Tel. 011/538166-7

L'editore non si assume responsabilità alcuna sul contenuto degli articoli e/o software inviati.
Per gli articoli firmati e/o siglati da collaboratori esterni, la redazione si assume la responsabilità prevista dalle leggi sulla stampa.

Tutti possono collaborare a TI 99 NEWSOFT, sia con articoli che con programmi. Scriveteci o telefonateci in redazione per avere tutti i ragguagli.

Spedizione in abb. post. Gr. III/70

AUTOBASIC CON LE CONTROL

Cogliamo al volo un ghiotto suggerimento di un nostro lettore, Giovanni Scopel, per proporvi i nostri lavori congiunti che riguardano l'uso del tutto nuovo di sfruttare i tasti di controllo della tastiera in modo da ottenere comandi ed istruzioni basic interpretabili ed eseguibili dal computer. La scoperta fatta si è rivelata, comunque, in tutt'e due i casi, assolutamente fortuita.
Non vogliamo più tenervi sulle spine, passiamo ai fatti. Accendete il computer, selezionate il linguaggio TI basic, e battere: 100 REM; poi, con tutta calma, premete control (CTRL) e insieme la lettera U. Concludete l'operazione con il solito ENTER. Date ora il LIST. Che cosa è successo? Al posto della control c'è il comando RANDOMIZE. Per verificare il suo funzionamento togliete la REM e provate a fare un programmino di estrazione casuale di numeri interi, aggiungendo queste tre linee:

```
110 A = INT (RND*100) +1
120 PRINT A
130 GOTO 100
```

Date ora il RUN e vedrete che, effettivamente, le cose vanno come devono andare. Bloccate pure il programma con FCTN 4 e imparate la trasformazione control-comando secondo questa tabella:

TASTO+ CTRL	FUNZIONE
1	TO
2	STEP
3	, (virgola)
4	; (punto e virgola)
5	: (due punti)
6	) (parentesi chiusa)
7	((parentesi aperta)
8	OPTION
9	OPEN
0 (zero)	THEN
=	CALL
Q	UNTRACE
W	READ
E	GO
R	INPUT
T	RESTORE
Y	DELETE
U	RANDOMIZE
I	DEF
O	UNBREAK
P	TRACE
/	... (AND)
A	ELSE
S	DATA
D	IF
F	GOTO
G	GOSUB
H	RETURN
J	DIM
K	END
L	FOR
;	PRINT
Z	REM
X	STOP
C	... (!)
V	NEXT
B	... (: :)
N	BREAK
M	LET
,	... (02 "+)
.	ON

Mentre nel TI Basic è comunque necessario mettere il comando REM prima della control per non avere il solito * BAD NAME, in extended Basic ciò è ammissibile a patto

però che non si aggiunga nessun altro carattere dopo la suddetta control altrimenti...
* ***UNRECOGINIZED CHARACTER.***

Da notare poi che mentre in TI BASIC le CTRL/CRTL C e CTRL B danno solo caratteri e simboli senza senso, sempre in extendend, equivalgono ai comandi AND, ! (commento), : : (elemento separatore tra due istruzioni). La CTRL, (virgola) non ha invece alcun effetto pratico in nessuno dei due linguaggi se non indecifrabile 02"+ in extended.
Si consiglia inoltre di utilizzare solo quei comandi veramente lunghi da digitare visto che aumentando il numero delle linee del programma il TI BASIC impiega anche più tempo a sostituirle. Riportati di seguito ci sono un paio di esempi su come utilizzare le control come comandi; a voi comunque la scelta per un migliore utilizzo delle stesse.

100 IF (A=34)+(B=12) THEN 100 ELSE 130 si può scrivere
100 REM (CRTL + D) (CTRL+7) A=34 (CTRL+6) (CTRL+ 7) B=12 (CTRL+6) (CTRL+0 (zero)) 110 (CTRL+A) 130
100 RANDOMIZE si può scrivere 100 REM (CTRL+U)

MARCO SQUINTANI

ANIMAZIONE DEGLI SPRITES

Un'importante caratteristica degli sprites è proprio quella di poterne cambiare la forma, senza che il programma ne risenta in termini di velocità. La magica istruzione è chiamata CALL PATTERN, e permette di variare la forma dello sprite da noi desiderato con quella specificata con il codice ascii all'interno della CALL.
Nell'ambito dei giochi, la tecnica di animazione più utilizzata dai programmatori è quella di definire una mole di caratteri all'inizio del programma per utilizzarli tramite call pattern per ottenere gli effetti grafici voluti.
Questa operazione anche se può sembrare molto valida in realtà limita le possibilità del gioco, infatti, i caratteri utilizzati per gli sprites, non potranno essere utilizzati per la grafica del paesaggio.
Se invece definiamo volta per volta solo i caratteri necessari per quella determinata fase di gioco, potremo certamente ottenere effetti e risultati migliori.

Un'altra tecnica efficace è quella di utilizzare figure composte da più di uno sprites, (per esempio un omino può essere diviso in due parti: gambe, resto del corpo), più il numero dei caratteri adibiti all'animazione sarà basso, più la velocità sarà maggiore.
Nel programma di esempio, possiamo notare come è stata organizzata la routine di animazione. L'animazione di «omini» è una tra le più complesse e richiede un numero di caratteri maggiore, in ogni caso siamo riusciti a contenerlo il più possibile, per tener fede a quanto detto sino ad ora.
Il programma è strutturato in blocchi, ognuno dei quali si occupa di animare l'omino per la direzione impostata.
Troviamo principalmente quattro fasi di movimento, alto con il tasto E, basso con il tasto X, sinistra e movimento, con il tasto S, destra e movimento, con il tasto D.
Nei sottoprogrammi di spostamento a sinistra e destra

utilizziamo una call locate per il movimento, perché una call motion causerebbe uno sfarfallio tra il carattere superiore e quello inferiore.
Il metodo di definire figure con molti sprites ci permette di diminuire ulteriormente il numero di caratteri da definire, infatti per ottenere lo stesso risultato, con un solo sprite, avremmo dovuto definire ben venti pattern anziché undici.
Il prossimo mese tratteremo ancora l'argomento animazione e vi spiegheremo come ottenere effetti grafici stupefacenti con la grafica multisprites.
Data la lunghezza del programma di esempio, abbiamo pensato di non farvi faticare troppo, e troverete il programma sulla cassetta.

EZIO MONTINI

```
1 !-----------------------------
2 !ANIMAZIONE SPRITE ESEMPIO
3 !            BY
4 !       MONTINI EZIO
5 !            FOR
6 !  TI 99/4A NEW SOFTWARE
7 !-----------------------------
100 CALL CLEAR
110 FOR W=0 TO 14 :: CALL COLOR(W,16,1):: NEXT W
120 CALL SCREEN(5)
130 DISPLAY AT(1,1):"        TI 99 NEW SOFT"
140 DISPLAY AT(3,1):"  ANIMAZIONE DEGLI SPRITES"
150 CALL MAGNIFY(3)
160 V,V1=100 :: V2=120 :: V3=124
170 CALL CHAR(96,"00070705070607017F7F6F6F6F6F6F6F00E0E0C0C0E0E0E080FEFEF6F6F6F6F6
F6")
180 CALL CHAR(100,"37070606060606060606060606000000ECE0606060606060606060606000000
000")
190 CALL CHAR(104,"37070706060606060606060606000000ECE0E060606060606060000000000000
000")
200 CALL CHAR(108,"37070706060606060600000000000000ECE0E060606060606060606060000000
000")
210 CALL CHAR(112,"00070707070703070606060606060606060C040C080C080C0C0C0C0C0C0C0C
0C0")
220 CALL CHAR(116,"00070507030703070606060606060606060C0C0C0C0C080C0C0C0C0C0C0C0C
0C0")
230 CALL CHAR(124,"07070701010101010101010100000C0C0C0000000000000000000C0000000
000")
240 CALL CHAR(120,"07070700000000000000000003000000C0C0C080808080808080808080000000
000")
250 CALL CHAR(128,"07070701010101070505010101000000C0C0C080402020C000000000C0000000
000")
260 CALL CHAR(132,"07070701020404030000000003000000C0C0C080808080E0A0A0808080000000
000")
270 CALL CHAR(136,"00070707070701 3F3F373737373737 00E0E0E0E0E080FCFCECECECECECECE
CEC")
280 CALL SPRITE(#1,96,16,100,112,#2,100,16,116,112)
290 CALL KEY(0,K,S)
300 IF S=0 THEN 290
310 IF K=ASC("E")THEN CALL PATTERN(#1,136,#2,V):: GOSUB 360
320 IF K=ASC("X")THEN CALL PATTERN(#1,96,#2,V1):: GOSUB 400
330 IF K=ASC("S")THEN CALL PATTERN(#1,116,#2,V2):: GOSUB 440
340 IF K=ASC("D")THEN CALL PATTERN(#1,112,#2,V3):: GOSUB 520
350 GOTO 290
360 REM AVANTI FRONTE
370 V=V+4
380 IF V>108 THEN V=100
390 RETURN
400 REM AVANTI RETRO
410 V1=V1-4
420 IF V1<100 THEN V1=108
430 RETURN
440 REM AVANTI SINISTRA
450 V2=V2+12
460 CALL POSITION(#1,R,C,#2,R1,C1)
470 C1=C1-5 :: C=C-5
480 IF C<1 OR C1<1 THEN C,C1=255
490 CALL LOCATE(#1,R,C,#2,R1,C1)
500 IF V2>132 THEN V2=120
510 RETURN
520 REM AVANTI DESTRA
530 CALL POSITION(#1,R0,C0,#2,R01,C01)
540 C0=C0+5 :: C01=C01+5
550 IF C01>255 OR C0>255 THEN C0,C01=1
560 CALL LOCATE(#1,R0,C0,#2,R01,C01)
570 V3=V3+4
580 IF V3>128 THEN V3=124
590 RETURN
```

DISPOSITIVI DI INPUT-OUTPUT

LA TASTIERA

Eccoci al consueto appuntamento con i vari componenti del nostro fedele Texas TI 99.

Dopo averlo esplorato in lungo e in largo trattando dei suoi più remoti nascondigli, passeremo ora ad esaminarlo sotto una veste certamente più familiare: l'esterno. Il nostro «amico» si presenta bene con la sua «carrozzeria» in plastica nera ricoperta da una satinatura d'argento (quasi un frac) e con il suo gradevole design.

Si nota immediatamente l'alloggiamento per i moduli di Solid State Software posto sulla destra e la tastiera che si trova, invece, in basso, a sinistra. La tastiera, come il monitor o televisore, rientra nella categoria dei dispositivi di input/output, cioè di entrata-uscita, veri responsabili delle comunicazioni tra utente e computer, aprendo un dialogo interattivo.

In un futuro quanto mai vicino (visto che è già presente in Giappone e negli Stati Uniti) sarà possibile dialogare direttamente a voce con il proprio computer attraverso un microfono e un convertitore di onde sonore di codici binari. Per il momento, però, accontentiamoci della tastiera che è ancora, senza dubbio, il dispositivo universale per il trasferimento delle informazioni dall'uomo al calcolatore.

Osservandola da vicino la vediamo in stretta analogia con quella di una comune macchina da scrivere, con tanto di caratteri alfanumerici (lettere, simboli, numeri) ai quali si aggiungono alcuni tasti di controllo del cursore e di funzione. Per quanto concerne i caratteri alfabetici, la nostra tastiera, essendo il TI 99 di progettazione e realizzazione americana, ha i tasti organizzati secondo le loro norme dattilografiche. Ciò significa che seguono lo standard QWERTY e non QZERTY come in Italia.

Il termine, che all'apparenza può sembrare indecifrabile, indica solo la sequenza delle prime sei lettere sulla tastiera. I caratteri numerici sono invece posti nella fila più alta e vanno dall'uno allo zero più un tasto aggiuntivo che dà l'uguale. In tutto 48 tasti, ma quasi ad ognuno è abbinata più di una funzione da svolgere, che considereremo più avanti.

Al centro, verso sinistra, spiccano le quattro frecce di controllo del cursore sotto gli ormai arcinoti tasti E, S, D, X, che rendono più agevole il compito di correzione e di ripresa delle parole. Altre opzioni molto importanti presenti sulla nostra tastiera sono quelle di autorepeat, minuscole (a dopo le critiche) e di ALPHA LOCK. L'autorepeat permette di ripetere un carattere più volte mantenendo premuto il tasto prescelto ed è molto utile, per esempio, nell'impostare da program-

GIOCHIAMO CON IL TI 99/4A

Un libro che non deve assolutamente mancare nella vostra libreria.

Potrete riceverlo a casa vostra a sole **L. 9500** + Spese di spedizione.

Per ogni richiesta di contrassegno, utilizzate il tagliando qui sotto spedendolo a:

NEWSOFT snc
via S. Jacini, 4
20121 MILANO

Vogliate spedirmi in contrassegno di L. 9500 + spese il libro: «GIOCHIAMO CON IL TI 99/4A»

Cognome .....

Nome .....

Indirizzo .....

.....

Cap. ..... Loc. .....

ma lunghe file di 0 e di F delle CALL CHAR o di muovere rapidamente e con continuità l'omino di un arcade game o il pannello di una routine grafica.
Sulle minuscole sappiamo già di dover ammettere, come prima cosa, la notevole carenza della definizione grafica che le rende, più che altre, maiuscole compresse, ma — magra consolazione — almeno ci sono e sono ridefinibili (V. TI 99 New Soft n. 0). Per evitare la fastidiosa operazione di «shiftare» le lettere per ottenere dalle minuscole le maiuscole, è presente il tasto di ALPHA LOCK.
Le dimensioni dei tasti e la disposizione della tastiera, parlando da un punto di vista strettamente ergonomico, offrono una certa piacevolezza e praticità all'uso. Valga per tutte l'intelligente omogeneità dei tasti di funzione della prima riga in alto. Altro grande pregio della nostra tastiera, o Keyboard, che dir si voglia, è quella di avere dei vari tasti e non dei sottoprodotti gommosi (V. ZX Spectrum, Sega SC 700, Aquarius, ecc.) o a membrana (ZX 81). Sarebbe stato magnifico poter usare un solo tasto per definire un'intera parola-chiave del Basic — direte voi —. Ma qui a «TI 99 New Soft» non siamo mica gli ultimi allampanati dell'ospizio...
Ce l'abbiamo fatta, come potrete andare a constatare nella rubrica TI Basic di questo mese.
Vanagloria a parte, vi diamo appuntamento al prossimo numero dove passeremo a cosiderare un po' più tecnicamente la nostra bella tastiera. Per il momento, accontentatevi di controllare il vostro grado di apprendimento con il programma TEACH BIT, che, in via del tutto eccezionale, questo mese fa sfoggio di un nuova grafica in alta risoluzione.

MARCO SQUINTANI

COME OTTENERE IL TEXT-MODE

Il nostro computer basa le sue capacità grafiche sul microprocessore TMS 9918a. Esso è in grado di formattare il video in ben quattro modi grafici, oltre alla possibilità di poter gestire gli SPRITES.
Per poter funzionare deve avere a disposizione una determinata quantità di memoria, dove depositare i dati che riguardano il video.
A secondo che si stia lavorando in graphics mode o in un altro dei tre modi grafici rimanenti, la quantità di memoria necessaria varia notevolmente.
Si pensi che per poter gestire correttamente il Bit Map mode, di cui abbiamo sentito molto parlare, occorrono ben 14000 bytes.
I computer della generazione MSX dedicano al TMS 9918a ben 16000 bytes. Solamente in queste condizioni di memoria, è in grado di poter gestire i modi grafici.
La memoria dedicata al microprocessore sul nostro computer è di soli 6kbyte circa.
In queste condizioni, il TMS 9918a è costretto ad utilizzare solo il modo grafico denominato graphics I mode (32x24), ridotte drasticamente a (28x24) per ragioni di distorsione del video dei normali televisori.
Tuttavia è possibile per i possessori di espansione di memoria, usufruire dei diversi modi grafici, utilizzando l'espansione come buffer di dati video.
Prima di parlare di modi grafici, dobbiamo spiegare cosa sono i VDP registers. Detti registri hanno la funzione di controllare e comandare il TMS 9918a, permettendoci di ordinare al microprocessore grafico di eseguire uno dei modi grafici. I registri VDP hanno anche la funzione di gestire gli sprites. I registri dedicati a questo compito di supervisione sono 8 (da 0 a 7).
Ogni registro svolge una ben determinata funzione di comando. Addirittura ogni bit del registro svolge una funzione, ed è necessario quindi agire con la massima cautela e precisione, onde evitare che il computer vada in CRASH con conseguente perdita di dati e... di salute per l'operatore!!!.
Potete trovare una dettagliata mappa dei registri nella figura 1.
Per operare il cambiamento di modo grafico, è necessario operare su 3 bit, che hanno il nome di M1, M2, M3. A seconda che questi bit siano settati o meno, possiamo ottenere quattro combinazioni.
Se M1, M2=0 e M3=0 il modo grafico ottenuto sarà : GRAPHICS 1 (24x32 colonne).
Se M1, M2=0 e M3=1 il modo grafico ottenuto sarà : GRAPHICS II (192x256 punti).
Se M1, M3=0 e M2=1 il modo grafico ottenuto sarà : MULTICOLOR (24x64 colonne).
Se M2, M3=0 e M1=1 il modo grafico ottenuto sarà : TEXT (24,40 colonne).
In ogni caso, potete trovare una tabella molto dettagliata,

ABBONATEVI
A TI 99
NEWSOFT.
CON SOLE
L. 95.000
RICEVERETE
11 NUMERI
DELLA RIVISTA
E IN PIU'
AVRETE
DIRITTO AGLI
SCONTI SUL
SOFTWARE
E
L'HARDWARE
OFFERTO DA
TI 99 MARKET

Per abbonarsi
basta inviare
un assegno
non trasferibile
o un vaglia
postale a:

NEWSOFT SNC
Via S. Jacini, 4
20121 MILANO

SCHEMA DEI REGISTRI VDP

Reg. n.	Bit n.	DESCRIZIONE
0	0 - 5	Riservati per uso futuro
	6	Se uguale 1 BIT MAP M3
	7	Abilita o disabilita VDP
1	0	Selezione banchi memoria 4/16 kram di memoria 0 = 4kram 1 = 16kram
	1	Abilita o disabilita il video 0 = off 1 = on
	2	Abilita o disabilita interrupt 0 = off 1 = on
	3	Se uguale 1 TEXT MODE M1
	4	Se uguale 1 MULTICOLOR M2
	5	Bit riservato al sistema
	6	Grandezza sprites : 0 = 8x8 bit 1 = 16x16 bit
	7	Magnify sprites : 0 = 1x 1 = 2x
2	0 - 3	Bit riservato al sistema
	4 - 7	Contiene posizione STBA Tavola dello schermo
3	0 - 7	Contiene posizione CTBA Tavola dei colori
4	d - 4	Bit riservati al sistema
	5 - 7	Contiene posizione PDBA Tavola delle forme
5	0	Bit riservato al sistema
	i - 7	Contiene posizione SATBA Tavola velocita' sprites
6	0 - 4	Bit riservati al sistema
	5 - 7	Contiene posizione SPTBA Tavola forme sprites
7	0 - 3	Colore Foreground caratteri.
	4 - 7	Colore Background caratteri e colore screen

Fig. 1

SCHEMA DEI MODI GRAFICI			
M1	M2	M3	MODO GRAFICO OTTENUTO
0	0	0	GRAPHICS I MODE 32 COLONNE
0	0	1	GRAPHICS II MODE BIT-MAPPED
0	1	0	MULTICOLOR MODE 64 COLONNE
1	0	0	TEXT MODE 40 COLONNE

Fig. 2

per quanto riguarda M1, M2, M3 in figura 2.
Per poter ottenere il Text mode, come abbiamo detto, dobbiamo settare il bit M1 contenuto nel registro 1.
L'unico modo per poter scrivere in questi registri è di operare una VWTR, mettendo in R1 il numero del registro vdp su cui si vuole operare e di seguito il valore da inserire.
Nel MSB metteremo in questo caso >01, perché vogliamo operare sul reg. vdp 1 e nel LSB metteremo <F0 per ottenere il text mode.
Ovviamente nel programma text mode, accluso alla rubrica SOFTWARE EXT+32k, non esiste una routine in linguaggio macchina per tornare al modo grafico normale, perché tale passaggio viene effettuato automaticamente dall'interprete Basic quando il programma viene breccato.
Arrivederci alla prossima puntata con MULTICOLOR e BIT MAP MODE...

EZIO MONTINI

CONFIGURAZIONE DEL VDP R1 PER OTTENERE GRAPHICS I

0	1	2	3	4	5	6	7
1	1	1	0	0	0	0	0

CONFIGURAZIONE DEL VDP R1 PER OTTENERE TEXT MODE

0	1	2	3	4	5	6	7
1	1	1	1	0	0	0	0

3 NUOVE ISTRUZIONI PER GESTIRE IL TEXT-MODE

Questo programma aggiunge tre istruzioni al normale Extended Basic, permettendo l'utilizzo, anche se in modo un po' complesso, del text mode.
Il text mode permette di utilizzare 40 colonne al posto delle normali 32, però presenta delle limitazioni, ad esempio non è possibile utilizzare gli sprites, così come non è possibile variare il colore dei caratteri.
Tutto questo ci fa pensare «a cosa serve allora?». La risposta e quanto mai semplice; infatti il text mode, come dice il nome, è stato ideato per l'elaborazione di testi. Noi per darvi l'opportunità di poterlo apprezzare, abbiamo creato un programma in linguaggio macchina in grado di poterlo utilizzare.

Dobbiamo tenere conto del fatto che tutte le istruzioni EXT., quali accept, display, hchar ecc. sono state create per funzionare in ambito 32 colonne, quindi non stupitevi se facendo una input, vi troverete il cursore quasi nel centro del video.
Per ovviare a tale mancanza, un sottoprogramma assembly vi permette di scrivere in qualunque parte dello schermo.
Il programma da noi ideato non pretende certo di essere perfetto, anzi presenta alcune limitazioni, per cui saranno ben accette le vostre opere di miglioramento, e speditele a TI 99 NEW SOFT, allegando una attenta e documentata spiegazione.
Tornando al TEXT MODE, avevamo detto che ci sono tre nuove istruzioni:

1) CALL LINK («MODE»)

Essa si occupa di convertire il formato del video da 32 a 40 colonne. Detto cambiamento causa la comparsa di sporcizia nelle ultime 5 righe, quindi si è reso necessario l'adozione di un sottoprogramma che pulisse tale area.

2) CALL LINK («CLEAR»)

Dal nome è facile dedurre la funzione di tale comando.
Essa si occupa di pulire lo schermo, composto da 960 locazioni, molto simile a CALL CLEAR. Ricordatevi che CALL CLEAR non cancella la sporcizia, perché il suo compito è di pulire solo 768 locazioni e non 960.

3A) CALL LOAD (9000, CA, CA, R1, R2)

3B) CALL LINK («PRINT»)
Questa istruzione è un misto di due istruzioni, infatti noi dobbiamo eseguire prima una CALL LOAD alla locazione dec. 9000 con quattro valori che rappresenta:

CA = codice ASCII del carattere da printare + offset 96.

R1 = intero della moltiplicazione Riga * Colonna diviso 256.

R2 = Riga * Colonna diviso 256 meno R1 moltiplicando il risultato per 256

In pratica R1 e R2 rappresentano la traduzione in NIBBLE della locazione dello schermo dove vogliamo sia printata la lettera da noi scelta.

Per chiarirvi le idee sull'argomento, pubblichiamo un programma che gestisce queste nuove istruzioni.

Per tornare al modo grafico normale basterà breccare il programma.

IMPORTANTE: prima di mettere in esecuzione il programma di esempio, assicuratevi di aver caricato, mandato in RUN e cancellato con NEW il programma che trovavate sulla cassetta col nome TEXT-MODE.

Detto programma può anche essere agganciato tramite un merge a programmi da voi creati.

EZIO MONTINI

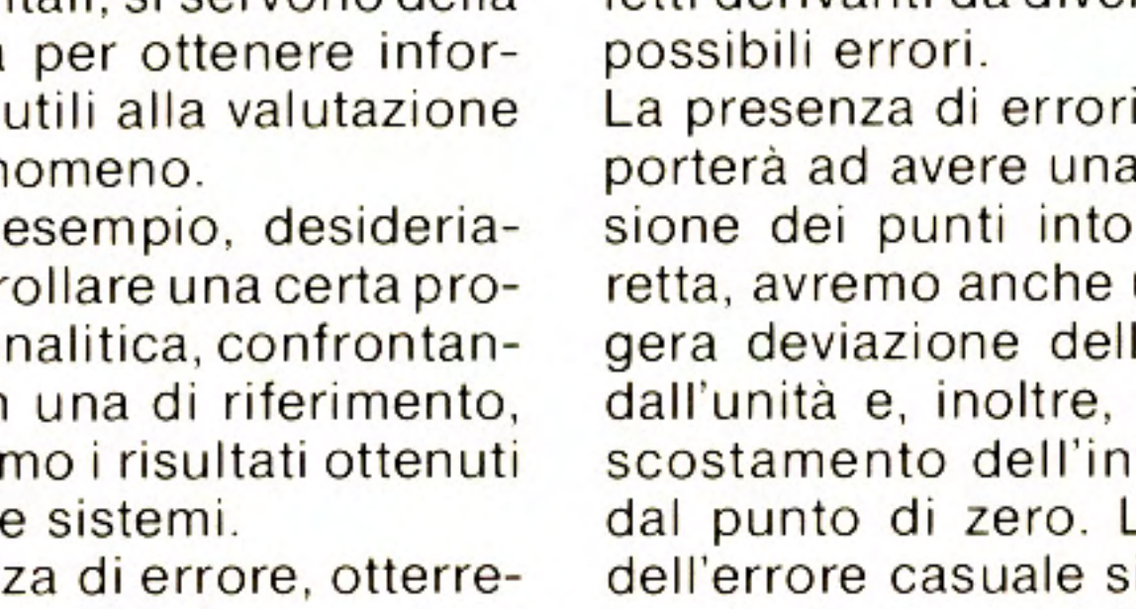
```
100 !+--------------------+
110 !! PROGRAMMA TEXT MODE :
120 !!         BY         :
130 !!    EZIO MONTINI    :
140 !!  Co.Mo.S  Software :
150 !+--------------------+
160 CALL INIT
170 CALL LOAD(16360,80,82,73,78,84,32,38,64)
180 CALL LOAD(16368,67,76,69,65,82,32,38,32)
190 CALL LOAD(16376,77,79,68,69,32,32,37,244)
200 CALL LOAD(8194,38,84,63,232)
210 CALL LOAD(9460,0,0,0,0,0,0,0,0,0,0,0,0,203,20,203,53,203,78,203,231,204,71)
220 CALL LOAD(9482,204,150,204,228,205,29,205,75,205,96,33,131,35,253,38,184,40,
183,41,182,42,195)
230 CALL LOAD(9504,43,193,44,179,45,194,47,196,58,181,59,180,60,191,61,190,62,19
2,94,197,255,58)
240 CALL LOAD(9526,58,130,65,84,240,71,79,133,73,70,132,79,78,155,79,82,186,80,7
3,221,84,79)
250 CALL LOAD(9548,177,255,65,66,83,203,65,76,76,236,65,78,68,187,65,83,67,220,6
5,84,78,204)
260 CALL LOAD(9570,66,89,69,3,67,79,78,1,67,79,83,205,68,69,70,137,68,73,77,138,
69,78)
270 CALL LOAD(9592,68,139,69,79,70,202,69,88,80,206,70,79,82,140,73,78,84,207,76
,69,78,213)
280 CALL LOAD(9614,76,69,84,141,76,79,71,208,77,65,88,223,77,73,78,224,78,69,87,
0,78,79)
290 CALL LOAD(9636,84,189,78,85,77,4,79,76,68,5,80,79,83,217,82,69,67,222,82,69,
77,154)
300 CALL LOAD(9658,82,69,83,6,82,78,68,215,82,85,78,169,83,71,78,209,83,73,78,21
0,83,81)
310 CALL LOAD(9680,82,211,83,85,66,161,84,65,66,252,84,65,78,212,86,65,76,218,88
,79,82,188)
320 CALL LOAD(9702,255,66,65,83,69,241,66,69,69,80,238,67,65,76,194,139,2,224,13
1,224,4,193)
330 CALL LOAD(9724,2,1,240,129,216,1,131,212,216,1,140,2,6,193,216,1,140,2,2,1,2
44,135)
340 CALL LOAD(9746,216,1,140,2,6,193,216,1,140,2,194,202,6,155,194,139,4,192,4,1
93,4,194)
350 CALL LOAD(9768,2,0,0,0,2,1,0,32,4,32,32,32,5,128,2,128,3,193,18,248,194,202)
360 CALL LOAD(9790,4,91,194,139,4,192,192,96,35,40,192,32,35,42,4,32,32,32,194,2
02,4,91)
370 CALL LOAD(9812,255,0)
```

```
100 '*************************
110 '*   ESEMPIO UTILIZZO    *
120 '* PROGRAMMA  TEXT-MODE *
130 '*          BY           *
140 '*     EZIO MONTINI      *
150 '*          FOR          *
160 '*  TI 99/4A NEW SOFT    *
170 '*************************
180 CALL CLEAR
190 '*************************
200 'METTE IN TEXT E PULISCE
210 '*************************
220 CALL LINK("MODE")
230 CALL LINK("CLEAR")
240 FOR A=1 TO 8
250 READ R1,C0,S$
260 '*************************
270 'CALCOLO LOC. VIDEO
280 !*************************
290 RX=((R1-1)*40)+C0-1
300 FOR L0=1 TO LEN(S$)
310 0$=SEG$(S$,L0,1)
320 CA=ASC(0$)+96 :: R=RX+L0
330 R1=INT(R/256)
340 R2=((R/256)-R1)*256
350 CALL LOAD(9000,CA,CA,R1,R2)
360 CALL LINK("PRINT")
370 NEXT L0
380 NEXT A
390 CALL KEY(0,K,S):: IF S=0 THEN 390 ELSE CALL CLEAR :: END
400 DATA 5,11,T E X T - M O D E,8,11,40  C O L O N N E,11,0,--------------------
--------------------
410 DATA 13,0,01234567890123456789012345678901234567890123456789,15,0,----------
----------
420 DATA 18,9,TI 99/4A  NEW SOFTWARE,20,6,SOFTWARE EXTENDED + 32 KRAM,24,6,PREMI
 UN TASTO PER TERMINARE
```

```
*********************************
*                               *
*    TEXT MODE  40 colonne      *
*              by               *
*         EZIO MONTINI          *
*                               *
*       Co.Mo.S SOFTWARE        *
*                               *
*********************************
*
* SORGENTE PROGRAMMA TEXT MODE
*
        DEF MODE,CLEAR,PRINT

PBUFF   BSS 256
SW      EQU >2020   * VSBW
MR      EQU >202C   * VMBR
MW      EQU >2024   * VMBW
BUFF    EQU 9000
BUFFR   EQU 9002

*
* SOTTOPROGRAMMA "MODE"
*

MODE    MOV  R11,R10
        LWPI >83E0
        CLR  R1
        LI   R1,>F081
        MOVB R1,@>83D4
        MOVB R1,@>8C02
        SWPB R1
        MOVB R1,@>8C02
        LI   R1,>F487
        MOVB R1,@>8C02
        SWPB R1
        MOVB R1,@>8C02
        MOV  R10,R11
        BL   *R11

*
* SOTTOPROGRAMMA "CLEAR"
*

CLEAR   MOV  R11,R10
        CLR  R0
        CLR  R1
        CLR  R2
        LI   R0,0
CONT    LI   R1,>20
        BLWP @SW
        INC  R0
        CI   R0,961
        JLE  CONT
        MOV  R10,R11
        B    *R11

*
* SOTTOPROGRAMMA "PRINT"
*

PRINT   MOV  R11,R10
        CLR  R0
        MOV  @BUFF,R1
        MOV  @BUFFR,R0
        BLWP @SW
        MOV  R10,R11
        B    *R11
        END
```

REGRESSIONE LINEARE

Il programma che questa volta vi presentiamo è scritto in TI Basic e tratta di un tipo di analisi essenziale in statistica e in tutte le discipline che, tramite l'osservazione di dati sperimentali, si servono della statistica per ottenere informazioni utili alla valutazione di un fenomeno.
Se, per esempio, desideriamo controllare una certa procedura analitica, confrontandola con una di riferimento, tracceremo i risultati ottenuti con i due sistemi.
In assenza di errore, otterremo una retta di equazione generale y = mx + c avente uno slope (o pendenza) m uguale a 1, una intercetta sull'asse delle ordinate, c, uguale a zero e tutti i punti osservati cadranno esattamente sulla retta.
Consideriamo adesso gli effetti derivanti da diversi tipi di possibili errori.
La presenza di errori casuali porterà ad avere una dispersione dei punti intorno alla retta, avremo anche una leggera deviazione dello slope dall'unità e, inoltre, un lieve scostamento dell'intercetta dal punto di zero. La stima dell'errore casuale si ottiene tramite il calcolo della deviazione standard nella direzione Y.



Retta raffigurante il caso ideale-tutti i punti giacciono esattamente sulla retta.

Un errore proporzionale sistematico causerà una variazione tangibile del coefficiente m per cui la differenza tra m e l'unità costituirà una stima di esso.
L'esistenza di un errore costante è, invece, dimostrata dallo scostamento del valore dell'intercetta dallo zero.

Possiamo pertanto affermare che lo studio della retta di regressione costituisce uno dei più potenti sistemi disponibili per l'analisi dei tre tipi di errore descritti, ed è un fondamentale sistema statistico per la comparazione dei dati.

Una parola d'avvertimento.

Detto questo dobbiamo ricordare che il problema con i calcolatori è costituito dal fatto che essi non sono dotati di senso critico. Un elaboratore che sia stato istruito a calcolare una retta di regressione fonirà SEMPRE come risultato la migliore linea passante per i punti dati e ciò anche nel caso questi rappresentino un cerchio (non oso pensare a quello che salterebbe fuori in questo frangente).

Avete acquistato il Numero di Maggio?

Ma sapete che sono usciti i Numeri di Giugno, Settembre, Ottobre, Novembre, Dicembre, Gennaio, Febbraio, Marzo e Aprile?

Per riceverli direttamente a casa vostra, scriveteci!

Ve li invieremo in contrassegno a L. 15000

(+spese cadauno).

TI 99 NEWSOFT
Via S. Jacini, 4
20121 MILANO

Il programma allegato vi darà pertanto alcuni indicatori per stimare l'attendibilità del risultato.
Essi sono:
1) Il coefficiente di correlazione di Pearson r.
2) La deviazione standard dei punti della retta.
3) Gli errori singoli tra ogni valore Y osservato e quello calcolato sulla retta.
4) La somma degli errori quadratici.
Analizziamo praticamente queste informazioni:
Il coefficiente r dovrà essere, per una retta accettabile, compreso tra 0.95 e -0.95.
Nel caso, come prima accennato, il calcolatore abbia tentato la quadratura del cerchio, si otterrà per r un valore di zero, o molto prossimo a zero, e sapremo, quindi, che l'analisi non è possibile.
Il coefficiente di Pearson non sarà, purtroppo, di grande aiuto nel caso in cui la curva sia appena accennata. La seconda barriera è costituita dalla deviazione standard.
È implicito che un alto valore della deviazione standard può significare uno scarso accordo dei dati. Purtroppo questo valore è, per sua stessa natura, legato anche alla grandezza del valore Y (un aumento di questa quantità corrisponde a un analogo aumento della sua deviazione standard).

Resta pertanto a voi decidere, caso per caso, l'importanza relativa di questa informazione. Il parametro degli errori singoli è quello che ci dà le informazioni più esaurienti sulla presenza di punti del tutto estranei alla retta calcolata oppure se, per qualche motivo, essa è soltanto una sommaria interpolazione di un andamento curvilineo.

Vi sarà facile con questo programma eliminare, come prova, alcuni di questi punti anomali e decidere in base ai risultati ottenuti, se la retta è una verosimile descrizione del fenomeno.

Fate però attenzione; qualsiasi eliminazione di dati sperimentali costituisce un, per così dire, travisamento dei dati storici: non vorremmo che, in ultima analisi, siate proprio voi a tentare la famosa quadratura del cerchio.

Dovremmo accennare alla verifica dello slope, e al limite di confidenza per il valore di Y, ma siamo certi, avendo già introdotto su queste pagine il test T e F, che sarete in grado, impiegando i programmi già pubblicati, di eseguire questi calcoli.

PAOLO CIVARDI

COME COSTRUIRE UN ARCADE GAME: SPECIALE LABIRINTI

Questo mese parliamo di labirinti.
Dopo quelli spaziali, i labirinti sono i giochi più apprezzati.
Il capostipite di questo genere di giochi, è il PAC-MAN della BALLY. Ora, nel nostro articolo, non vogliamo spiegarvi né come si gioca, né che cosa sono i labirinti, vogliamo insegnarvi a costruirli.
Normalmente, nei giochi, i labirinti sono ciclici, cioé dopo 3 o 4 schemi, riprendono ad apparirci, a cominciare dal primo e cosi via.
Sarebbe invece interessante sfruttare le capacità del vostro computer e generare labirinti sempre diversi. Per potere costruire un labirinto, per prima cosa è necessario stabilire dove si entra e dove si esce: stabiliti questi due punti, si deve costruire o meglio collegare questi due punti con una strada la più tortuosa possibile, quindi tracciare dei vicoli ciechi o dei giri viziosi che incrocino tale tracciato in modo da confondere le idee a chi gioca.
Questa è grosso modo l'idea che svilupperemo questo mese anche perché molti ci hanno scritto dicendo che parliamo sempre di giochi spaziali.

A questo proposito, teniamo a smentire tutti coloro che sono della stessa idea, poiché gli articoli finora pubblicati erano di interesse generico e quindi applicabili in qualunque contesto, anche ai LABIRINTI.

Naturalmente le idee sui labirinti possono essere molteplici, noi come al solito ne suggeriamo alcune, mentre le restanti, partendo dal nostro spunto, le analizzerete da voi.

Per sviluppare la nostra idea è necessario dimensionare una matrice che dovrà contenere il nostro tracciato, e tutto ciò che vorremo inserire all'interno del nostro labirinto.
Per potere toccare con mano la cosa, partendo dalle routines che alleghiamo all'articolo, abbiamo sviluppato un super ADVENTURE grafico, con una mappa che cambia ad ogni nuova partita. Ed è visualizzata in 3-D; quindi le routines che seguiranno le vedrete all'opera nel programma.
Stabiliamo ora la grandezza della mappa in righe e colonne; queste righe e colonne serviranno a dimensionare la matrice a 3 elementi M$ (R,C,2) dove «R» rappresenta la lunghezza del lato del labirinto; «C» rappresenta la lunghezza del lato corto e «2» sono i nostri contenitori:
>1) Contiene il percorso
>2) Contiene gli oggetti che troviamo nel labirinto
Per comoditàa abbiamo assunto R=*20 e C=10 quindi:*
>17 DIM M$ (20, 10, 2)
Il motivo per cui la matrice è di stringa, dipende dal fatto

che dobbiamo occupare il minore numero di bytes possibili, poiché nella matrice inseriremo solo numeri da 1 a 9; ogni numero nella matrice di stringa occuperà 1 byte, mentre in quella numerica avrebbe occupato 16 bytes.
Ritornando al labirinto, abbiamo stabilito che si tratterà di tante stanze contigue di ben 4 tipi:

>1) Stanza chiusa su 4 lati
>2) Stanza aperta su 4 lati
>3) Percorso più breve o tracciato principale
>4) Uscite

A questo punto occorre trovare un algoritmo in grado di creare una traccia coerente in grado di portarci da un estremo all'altro del labirinto, e riempire queste celle di «3».
Vediamo un esempio:

Come potete vedere nell'esempio, abbiamo ottenuto una mappa che contiene, ad ogni sua posizione, un numero da 1 a 4 che serve a distinguere il tipo di stanza.
Per ottenere questo tipo di mappa si è ricorsi alla «RANDOMIZE» ottenendo così la possibilità di avere labirinti sempre diversi tra loro ad ogni partita.
Vediamo allora di capire il funzionamento della routine da noi creata allo scopo:

```
>2 RANDOMIZE
```

Il comando serve per ottenere un randomize non ciclico. Poiché inoltre deve essere utilizzata diverse volte nel programma, si è pensato di ricorrere allo statement:

```
>4 DEF U= INT (RND*S)
```

otteniamo cosi la possibilità di avere il randomize ogni qualvolta ne abbiamo bisogno, definendo qualora c'è ne fosse bisogno, la variabile «S» e richiamando la «U». Ed ecco ora la routine passo passo:

```
>389 N=1
>390 S=7
>391 X=U+3
>392 S=4
>393 Y=U+2
>394 FOR I=N TO N+Y
>395 IF I=19 THEN 415
>396 M$ (I, X, 1) ="3"
>397 NEXT I
```

In questa prima parte «N» viene messo ad 1 per indicare che la colonna di partenza sarà la colonna 1; poi viene randomizzato un valore da 3 a 7 come riga di partenza, nell'esempio «X» vale 7, quindi la «Y» assumerà il valore, sempre randomizzato, della lunghezza del primo tratto di corridoio, espresso in stanze, quindi con il ciclo FOR-NEXT in dette stanze, viene inserito il valore «3».
La riga 395 è stata messa poiché il ciclo viene ripetuto tante volte finchè il valore di «N» non sarà pari a 19 cioè pari alla lunghezza del labirinto.

```
>398 N=N+Y
>399 M$ (N+1, X, 1)= "1"
>400 S=2
>401 K=U+1
>402 IF K <> 2 THEN 404
>403 K=-1
>404 S=3
>405 S=U+2
>406 FOR I=1 TO S
```

```
---------+---------+---------+---------+---------+---------+---------+---------+---------+---------
!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!
!4!1!1!1!1!2!2!2!2!3!3!3!3!3!3!3!3!3!2!1!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!1!
!4!1!3!3!3!3!1!1!2!3!2!2!2!2!2!1!2!2!2!1!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!1!
!4!1!3!1!1!3!1!1!2!3!2!2!1!2!1!2!1!1!2!1!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!1!
!4!1!3!2!1!3!3!3!3!3!1!1!2!1!1!1!2!1!2!1!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!1!
!4!1!3!1!2!1!1!1!2!2!2!2!1!1!1!1!2!1!2!1!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!1!
!4!3!3!1!1!1!1!2!2!2!2!2!2!2!1!2!1!1!2!1!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!1!
!4!1!2!1!1!1!1!2!2!2!2!2!2!1!1!1!1!2!2!1!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!1!
!4!1!2!2!2!1!1!1!2!2!2!2!2!1!1!2!1!2!2!1!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!2!1!
!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!1!
---------+---------+---------+---------+---------+---------+---------+---------+---------+---------
```

```
>407 IF (X+K*I) <9 THEN
409
>408 X=X-1
>409 IF (X+K*I) >1 THEN
411
>410 X=X+1
>411 M$(N, X+K*I, 1)="3"
>412 NEXT I
```

Anche in questo caso vengono randomizzati dei valori indicanti le coordinate delle stanzed, ma a differenza della precedente routine, si cammina in senso ortogonale alla direzione precedente; anche qui delle «IF» controllano che non si esca dalla mappa.

```
>413 X=X+S*K
>414 GOTO 392
```

Chiudiamo la soubroutine con la riga di aggiornamento dei valori ed il ritorno a rieseguire la procedura fino al rimpimento della traccia principale; a questo punto si torna con un «RETURN» e quindi con la routine che segue, il programma provvede a completare il riempimento della matrice con dei «2» e «1».

```
>263 FOR I=1 TO LN
>264 FOR @=1 TO 10
>265 S=1.9
>266 IF M$(I,@,1)<>"" THEN 268
>267 M$(I,@,1)=STR$(U+1)
>268 NEXT @
>269 NEXT I
>270 FOR I=1 TO 10
>271 M$(LN,I,1)="1"
>272 M$(LN-1,I,1)="2"
>273 M$(1,I,1)="4"
>274 IF M$(2,I,1)="3" THEN 276
>275 M$(2,I,1)="1"
>276 NEXT I
>277 FOR I=1 TO LN
>278 M$(I,1,1)="1"
>279 M$(I,10,1)="1"
>280 NEXT I
```

Ora non manca che una routine che vi permetta la visualizzazione del labirinto creato dal vostro TI 99/4A e cosi esaudiamo anche questo vostro desiderio prelevando un'altra delle numerose routine dal programma

```
>102 RESTORE 557
>103 CALL CLEAR
>104 CALL COLOR(9,2,10)
>105 CALL COLOR(10,2,15)
>106 FOR @=102 TO 106
>107 READ A
>108 CALL CHAR(@,B$)
>109 NEXT @
```

```
>110 PRINT TAB(5);"QUESTA E LA PIANTA"
>111 P$=""
>112 FOR @=1 TO 10
>113 FOR I=1 TO LN
>114 IF (@=C)*(I=R) THEN 117
>115 B$=B$&CHR$(101+VAL(M$(I,@,1)))
>116 GOTO 118
>117 B$=B$&CHR$(106)
>118 NEXT I
>119 PRINT B$
>120 B$=""
>121 NEXT @
>122 PRINT
>123 A=24
>124 C=3
>125 FOR I=1 TO 5
>126 READ P$
>127 C=C+3
>128 GOSUB 426
>129 NEXT I
>130 FOR C=1 TO 1000
>131 NEXT C
>132 CALL CLEAR
>133 END
```

Quindi le linee data e la subroutine richiamata alla linea 128

```
>557 DATA 007E7E7E7E7E7E,E7818100000181E7,E7818100008181E7,E7818100008181E7,FF5757575757B1FFFF,E781A51818A581E7
>558 DATA f:CIECA,g:STANZA,h:USCITA,i:F.LAB.,j:GIOCAT.

>426 FOR @=1 TO LEN(P$)
>427 CALL HCHAR(C,A+@,ASC(SEG$(P$,@,1)))
>428 NEXT @
>429 RETURN
```

Con questo possiamo dire che per questo mese avete abbastanza «carne al fuoco (di drago!!!)».

MAURO MATTIAZZI (MMG)

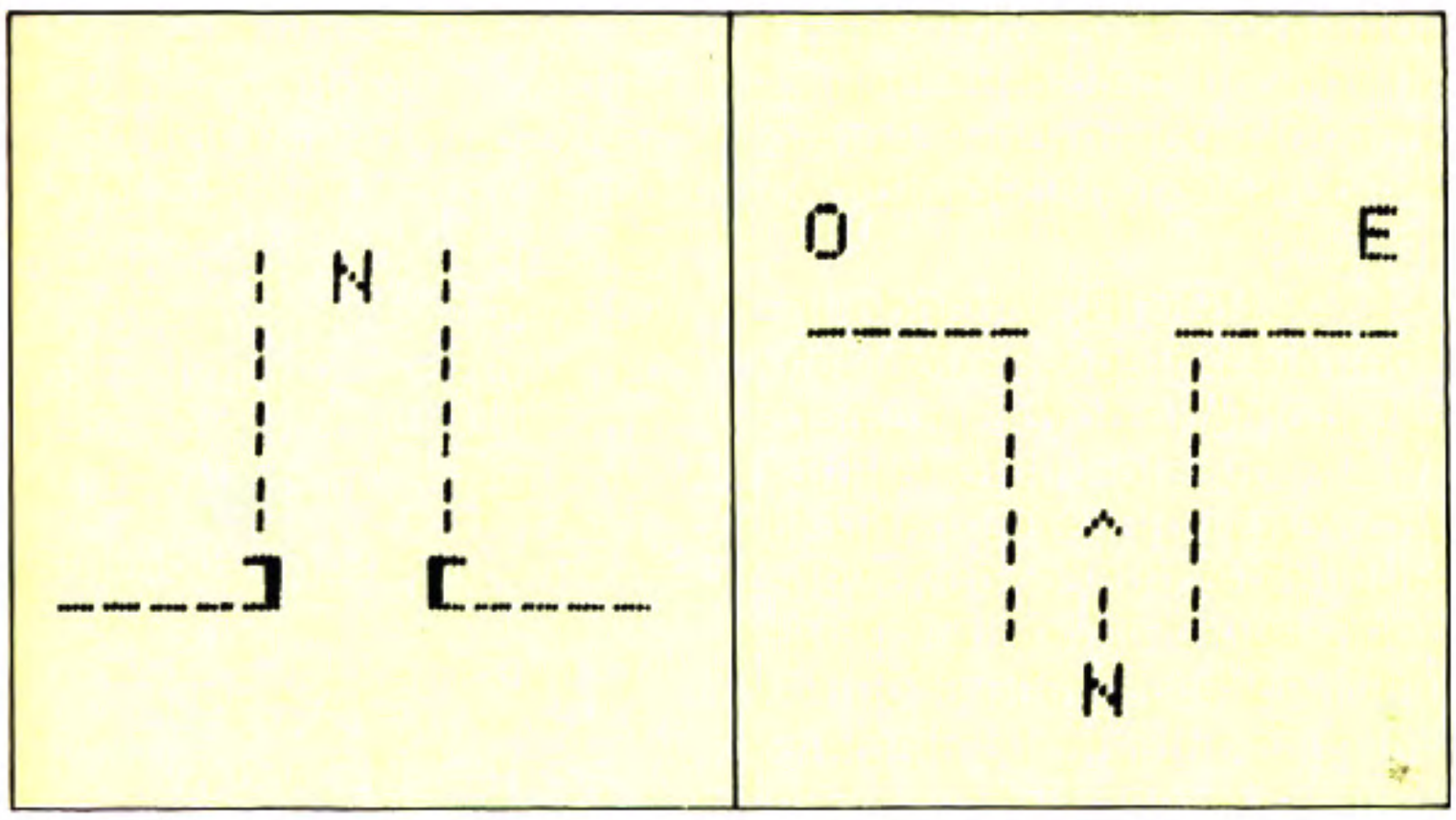
3-D MONSTERS MANOR

Ed eccoci a darvi le dritte per il gioco di questo mese. Dobbiamo dire che questa volta abbiamo fatto uno sforzo particolare, che va a vantaggio di coloro che hanno la macchina in configurazione minima, cioè in TI-BASIC, quindi accessibile a tutti.
Dobbiamo dire che qui alla M.M.G. siamo tutti soddisfatti del lavoro svolto, e ci teniamo a ringraziare tutti coloro che ci scrivono. Vogliamo inoltre puntualizzare su queste pagine, a coloro che ci richiedono il catalogo giochi, oppure giochi ben precisi, che siamo nell'impossibilità di accontentare tutti, e che presto i nostri giochi più belli potranno essere acquistati presso il TI 99 MARKET.
Infine vogliamo fare qualche anticipazione: consigliamo vivamente l'acquisto da parte dei nostri lettori del KIT TI-LOGO+Memory espansion 32K, poiché qui alla M.M.G. alcuni programmatori sono gia al lavoro per offrirvi grosse novità in logo che nei prossimi mesi potrete vedere sulle pagine di TI 99 NEWSOFT.
A questo punto torniamo al nostro gioco.
Prima della spiegazione una raccomandazione: se in teoria lo si potrebbe giocare in extended, vi consigliamo caldamente di usarlo esclusivamente in TI-BASIC, lo stesso vale per chi possieda l'espansione di memoria.
Dobbiamo dire che il gioco in realtà supera di gran lunga le capacità della macchina, e questo ci ha obbligati a fare un programma che opportunamente compattato, sfruttasse il minor numero di variabili possibili; cosi due sono i tipi di variabili; quelle del primo tipo sono i contenitori del gioco, cioè la matrice del labirinto, il portatore degli oggetti, i punteggi e cosi via, mentre quelle di secondo tipo sono le variabili ricorsive, cioè quelle che ad ogni routine vengono ridefinite ed utilizzate a scopi diversificati.
Di variabili ricorsive ne sono state adoperate 9, questo ha permesso di ridurre l'occupazione di memoria, ma ha reso il programma illeggibile.
Inoltre la struttura totalmente a sub-routines e tutti i DATA al fondo del programma, non permettono la lettura delle funzioni espletate dalle varie routines. Questa struttura tuttavia ha permesso la realizzazione del gioco, ed una velocità di esecuzione accettabile. Inoltre la memoria residua (molto poca) non ha permesso l'inserimento di REM.

IL GIOCO

Scopo del gioco è uscire dal labirinto nel minor numero di mosse possibile, con il maggior numero di «$» possibili.

I comandi sono segnati a lato della finestra video, mentre sotto i comandi sono visibili il contatore delle mosse, il contatore delle vite rimaste, ne avete 3 a disposizione, ed il contatore dei soldi «$». Inoltre sotto, mano mano che vi verrà data la direzione nella quale state guardando, vedrete le iniziali dei quattro punti cardinali. Fig. 1 - esempio punto di vista:



È da notare che le mosse verranno contate solamente quando premete «A» cioè avanti, mentre cambiando direzione, si effettua una rotazione su voi stessi che il computer non calcola come mossa. Infine a fianco del punto di vista c'è una spia quadrata: la spia serve per sapere quando potete premere i tasti comando. Infatti, poichè il lavoro affidato al computer è molto, durante i 4-5 secondi che il TI 99 lavora, la spia resta rossa (chiara per chi ha il B&N) e tutti i comandi digitati durante questo periodo vengono ignorati dalla macchina. Quando invece la spia è verde (grigio scuro in B&N) potete digitare il comando voluto.
A questo punto, tralasciando i ben noti comandi direzionali, vediamo di capire il gioco.
La prima cosa che il TI 99 chiede è se volete un gioco nuovo o uno vecchio. Se volete, una nuova partita premete «1» viceversa, se avete una partita da finire, memorizzata sul registratore, (vedremo poi come) premete il «2».
Tolto che premendo il «2» si avrà la solita procedura con il registratore, nel caso scegliete «1» il computer vi chiederà il livello di difficoltà del gioco: sono a vostra disposizione ben 15 livelli di difficoltà crescente. A questo punto, dopo che la macchina avrà creato la nuova mappa di gioco, verrà costruito il video di gioco, come visto in precedenza, e quindi iniziate a giocare.
La prima cosa che salta agli occhi, è la strana abbondanza di comandi! Dovete sapere che lungo il tragitto troverete degli oggetti che vi serviranno durante il cammino, dei pericoli e quindi tali comandi servono per risolvere queste situazioni.

OGGETTI

—) la BOTTIGLIA: serve a spegnere il fuoco.
—) la SPADA: serve ad aumentare il tuo fattore di combattimento. Più spade hai e più sono le probabilità in tuo favore di vincere il duello.
—) il FORZIERE: contiene il tesoro, ma per aprirlo è necessaria la chiave.
—) la CHIAVE D'ORO: serve ad aprire il forziere.
—) la CHIAVE D'ARGENTO: serve ad aprire la porta d'uscita; senza la chiave non è possibile uscire
—) i DIAMANTI: servono a comprare le spade.

PERICOLI

—) il DRAGO: il drago vi sbarra la strada. Potete affrontarlo oppure cambiare direzione, tenete presente che avete probabilità di riuscire a girargli intorno, come no, e può darsi che dietro il drago ci siano oggetti di vitale importanza. Il fattore combattimento del drago è in funzione del vostro; senza spade il vostro F.C. è uguale a 15 mentre quello del drago è 85; ogni spada vi aumenta il F.C. di 10 mentre l'F.C. del drago diminuisce di conseguenza.
—) la BOTOLA: se cadete nella botola venite portati dal passaggio botola in un'altra stanza del labirinto
—) il FUOCO: vi sbarra la strada, potete spegnerlo solo nel caso abbiate una bottiglia: ricordate che una bottiglia serve una volta sola.
È inoltre da tenere presente che potete portare massimo 10 oggetti.

COMANDI DEDICATI

—) F : FUORI, questo comando serve a chiudere il gioco. Premendo «F» il programma vi mostra la pianta del labirinto e la vostra posizione al suo interno.
—) Q : CAMBIO, è il comando di emergenza. Se vi accorgete che siete imbottigliati premete «Q» e verrete teleportati in un'altra zona del labirinto. Può essere usato solo 2 volte.
—) J : SPEGNI, serve a spegnere il fuoco.
—) P : PRENDI, serve a prendere gli oggetti (max 10)
—) R : CS1, serve nel caso vogliate smettere di giocare ma volete finire la partita, con la situazione attuale con questo comando, è possibile registrarla su cassetta richiamandola poi in un secondo tempo selezionando il gioco vecchio.
—) U : UCCIDI, quando incontrate il drago, se decidete di affrontarlo dovete premere «U». A questo punto la finestra verrà chiusa e apparirà la tabella del duello, dove vengono aggiornati tutti i parametri necessari allo scontro.
Nel caso abbiate dei diamanti, vi verrà chiesto se volete acquistare delle spade. Passata la fase di transazione, inizia il duello vero e proprio.
Voi avete a disposizione 3 vite, il drago 1. Nel caso riusciate ad uccidere il drago potrete proseguire, viceversa, vi verrà mostrata la mappa quindi la partita avrà termine.

A questo punto possiamo dire di avervi detto tutto sul gioco e prima di salutarvi vi lanciamo una sfida: uscire dal livello 15 in meno di 40 mosse.

Secondo noi è una sfida interessante, provate e vedremo chi ci riesce.

MAURO MATTIAZZI
(MMG)

DICHIARAZIONE DEI REDDITI 1984

E' disponibile un programma
che finalmente vi guiderà alla determinazione
del reddito per i Mod. 101 e 740.

Il programma gira in Extended Basic
e costa Lit. 35.000 + spese di spedizione.

PER RICHIEDERLO
UTILIZZATE LA CARTOLINA
TI 99 MARKET

SNOW TREK

Programma: «**Snow Trek**» della Pewterware.
Configurazione richiesta: **consolle + registratore + cavo.**
Linguaggio: **Ti Basic.**

IL GIOCO

Dopo una breve presentazione, accompagnata sul finire delle prime note della canzone «Jingle Bells», appare un lungo messaggio in inglese che ambienta la situazione.
A causa di una tremenda bufera invernale la scuola è stata chiusa e voi, preso il gatto delle nevi, scorrazzate per tutto il paese coperto da una bianca coltre. La strada è però ghiacciata e dovete stare attenti a non andare a «sbattere», controllando lo slittino con i tasti 3, 4, 6 e 7 come da istruzioni.
Si passa ora alla descrizione del primo gioco, dove viene richiesto il vostro grado di abilità nel guidare lo slittino. Potrete scegliere tra: 1) mai guidato prima; 2) ho appena avuto la patente; 3) buon giudatore; 4) eccellente guidatore; 5) iscritto alla 500 miglia di Indianapolis. In caso di scarsa memoria sarà possibile tornare allo schermo delle istruzioni premendo lo 0. Ha inizio finalmente la vostra corsa, tesa, più che altro, a evitare pini ed abeti. Se ne urterete uno, infatti, il computer vi fornirà il punteggio e il record chiedendo inoltre se avete ancora voglia di giocare. In caso affermativo, premete la Y; altrimenti la N.
Usciti invece indenni dalla discesa, dovrete vedervela, tra steccati, laghetti e pini, con dei malefici numerini vaganti che dovrete acciuffare prima di terminare il carburante. I numerini in questione sono piuttosto veloci e si muovono in modo da farvi urtare gli ostacoli, ponendo così fine alla vostra partita. Per muoversi, in questo caso, basta premere 4 per andare a sinistra, 5 per frenare e 6 per andare a destra.
Uscire vittoriosi significherà passare ad un livello superiore, ricominciando dalla corsa tra gli alberi.
Date quindi sfogo alla vostra abilità di tipo sciatorio, senza — per altro — rompervi l'osso del collo (come potrebbe magari accadere nella realtà). Pistaaa...

MARCO SQUINTANI

BOUNCER «GRAFICA 3 DIMENSIONI»

Vi troverete in un altro mondo, dove sopravvivere alle tremende minacce degli attacchi nemici è meno facile di quanto voi pensiate.
Bouncer fa parte di una nuova serie di giochi che sta prendendo rapidamente mercato, esso è caratterizzato da ottimi effetti a tre dimensioni. Molto simile al famosissimo gioco che tutti o quasi abbiamo visto nei bar: Qebert o Cubit. Lo scopo del gioco e di saltare su tutti i quadratini sparsi sullo schermo, senza farsi «infilzare» dalle tremende frecce avvelenate dei nemici crudeli e sanguinari, sempre in agguato.
In genere sul cubo che compare nel centro dello scermo, compare un punto di colore nero, se noi saltando riusciamo a prenderlo, potremo usufruire del bonus. Il bonus consiste nel permetterci di prendere anche i cubi posti agli estremi del campo di gioco dandoci il bonus di punti.
I livelli di difficoltà sono dodici, ma non sono selezionabili dall'utente, quindi, per arrivare al dodicesimo livello, dovrete sudare, ma non scoraggiatevi perché tutto è possibile.
All'inizio del gioco, disponiamo di quattro vite, ed il nostro simpatico BOUCER sarà posizionato in alto al centro dello schermo.
Assicuratevi di aver alzato il tasto ALPHA LOCK, poi il programma vi chiederà quale dei due joystick volete utilizzare.
Nel caso non abbiate i joystick, aggiungete al programma le righe sotto riportate.

I tasti da utilizzare, questa volta con l'ALPHA LOCK premuto sono:

S: per saltare a sinistra
D: per saltare a destra
W: per saltare in alto a sinistra
C: per saltare in basso a destra
Z: per saltare in basso a sinistra
R: per saltare in alto a destra

EZIO MONTINI

Righe da aggiungere per adattamento della tastiere al posto dei Joystick 1 e 2.

```
30000 SUB JOYST (N, X, Y)
30010 CALL KEY (O, K, S) : : IF S=O THEN X, Y=O : : SUBEXIT
30020 ! IF K=69 THEN Y=4 : : X=0 : : SUBEXIT
30030 ! IF K=88 THEN Y=-4 : : X=0 : : SUBEXIT
30040 IF K=68 THEN X=4 : : Y=0 : : SUBEXIT
30050 IF K=83 THEN X=4 : : Y=0 : : SUBEXIT
30060 IF K=87 THEN X=-4 : : Y=4 : : SUBEXIT
30070 IF K=82 THEN X, Y=4 : : SUBEXIT
30080 IF K=90 THEN X, Y=-4 : : SUBEXIT
30090 IF K=67 THEN X=4 : : Y=-4 : : SUBEXIT
30100 X, Y=0 : : SUBEND
```

ALBUM

DESCRIZIONE

Il programma serve ad un bambino in età prescolare per imparare le lettere dell'alfabeto italiano.
Nel programma sono memorizzate oltre 40 figure, con grafica multicolore molto simile a quella degli album da disegno destinati alle prime classi della scuola elementare; dopo la comparsa della schermata iniziale (un vulcano schematizzato dalle parole costituenti il titolo, che erutta lettere dell'alfabeto) e le istruzioni, premendo un tasto inizieranno a comparire i disegni, e l'utente dovrà individuare il tasto corrispondente alla lettera iniziale della figura per vedere la lettera, e di seguito la parola stampata con effetto «telescrivente», accanto al disegno.
Le prime linee del programma definiscono i caratteri dal 130 al 159 in elementi tipo «mosaico», e colorano i 4 sets corrispondenti (13, giallo, 14, rosso, 15, verde, 16, magenta). Questi caratteri verranno poi indirizzati sullo schermo mediante istruzioni «print». La particolarità di questa tecnica sta nel fatto che una volta ridefiniti e colorati questi caratteri sono direttamente richiamabili da tastiera mediante CONTROL+. Il tasto corrispondente, come altri computer in commercio. Ricordate che saranno visibili solo dopo il primo «RUN».
Nel caso di questo programma i caratteri sono uguali per i primi 3 sets, e simili per il quarto; ma è anche possibile definire caratteri standard da richiamare uguali a quelli di VIC., SPECTRUM, C. 64 (per gli invidiosi!) allo scopo di facilitare la conversione di programmi, oppure un set originale adatto ai propri scopi, riportandolo su una striscia da sovrapporre all'originale sul lato superiore del computer.
Il vantaggio di questa tecnica è in primo luogo quello di poter comporre l'immagine sullo schermo in un tempo molto breve, e poi l'enorme spazio di memoria risparmiato; infatti con un'istruzione «CALL CHAR» è possibile indirizzare un solo carattere o tutt'al più quelli seguenti se uguali, mentre con un'istruzione «print» (o display) si può collocare sullo schermo decine di caratteri per ogni linea, fino al completamento delle 4 linee, ed in più, utilizzando la tecnica di forzamento, aggiungere ben 2 righe di caratteri ogni linea, e ancora risparmiare gli spazi con l'istruzione «TAB», e la punteggiatura del print (: = a capo, ; = accostato, ecc.). E, quindi possibile con 2 o 3 linee «PRINT» comporre un'immagine che richiederebbe 25/30 linee di «CALL HCHAR» (con conseguente spreco di memoria) ed un simpatico effetto «sipario» (l'immagine scorrerà dal basso verso l'alto).
In T4I BASIC + MINIMEMORY è possibile tenere nella «RAM» dell'accessorio (comandi: SAVE MINIMEM e OLD MINIMEM) un programma costituito da più blocchi che definiscono centinaia di caratteri, richiamandolo quando è necessario (prima di digitare un programma) o anche dopo con il «MERGE», e cancellando ciò che non interessa.
Continuando con questa rivalutazione delle istruzioni «PRINT» ricordiamo «DISPLAY USING», che consente data la straordinaria velocità di esecuzione, di comporre intere schermate che si susseguono per programmi di animazione, intervallando con istruzioni «CALL PATTERN» o «CALL COLOR» secondo la tecnica illustrata sul numero di novembre.
Quindi: 1) *display using* per comporre la schermata; 2) *call pattern* o *call color* per simulare il movimento; 3) *goto* ad una linea di *display using* che componga la schermata successiva della direzione indicata dal comando con incremento o decremento del numero di linea.

SERGIO PERRONE

AIUTO ALLE DECISIONI: SCEGLIAMO UN'AUTOMOBILE!

Questo programma fornisce un valido aiuto quando si debba effettuare una scelta fra più elementi e questa dipende da molti parametri.
Nell'esempio proposto si tratta di effettuare una scelta fra automobili ed i parametri di scelta sono:

— PREZZO
— NUMERO DI PORTE
— POTENZA
— VELOCITA' MASSIMA
— CONSUMI
— PESO
— DIESEL

Nel programma sono contenuti i dati di quasi tutte le autovetture FIAT.
Ad esempio la istruzione 1010 descrive la PANDA 30 e contiene:

1010	DATA ÷ E, 01, Panda 30, 6.9, 3, 30, 115, 6.4, 650, 0
÷E	serve al programma per riconoscere il tipo di dati che seguono
01	è il numero progressivo
Panda 30	è la descrizione della autovettura
6.9	è il prezzo in milioni
3	è il numero di porte
30	è la potenza in cavalli
115	è la velocità massima in Km/h
6.4	è il consumo medio in l/100 km
650	è il peso in kg
0	è un indicatore che vale 1 quando si tratta di un'auto Diesel

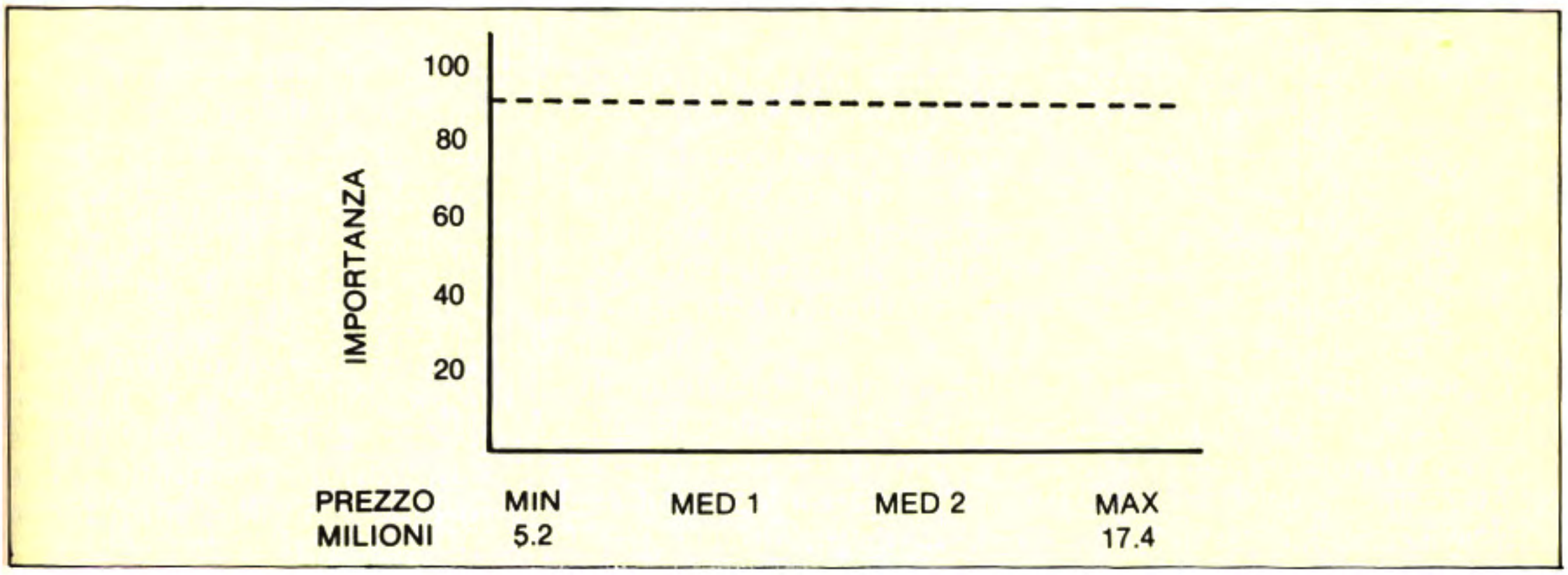
Nel programma possono essere descritte un massimo di 50 automobili.

RUN!

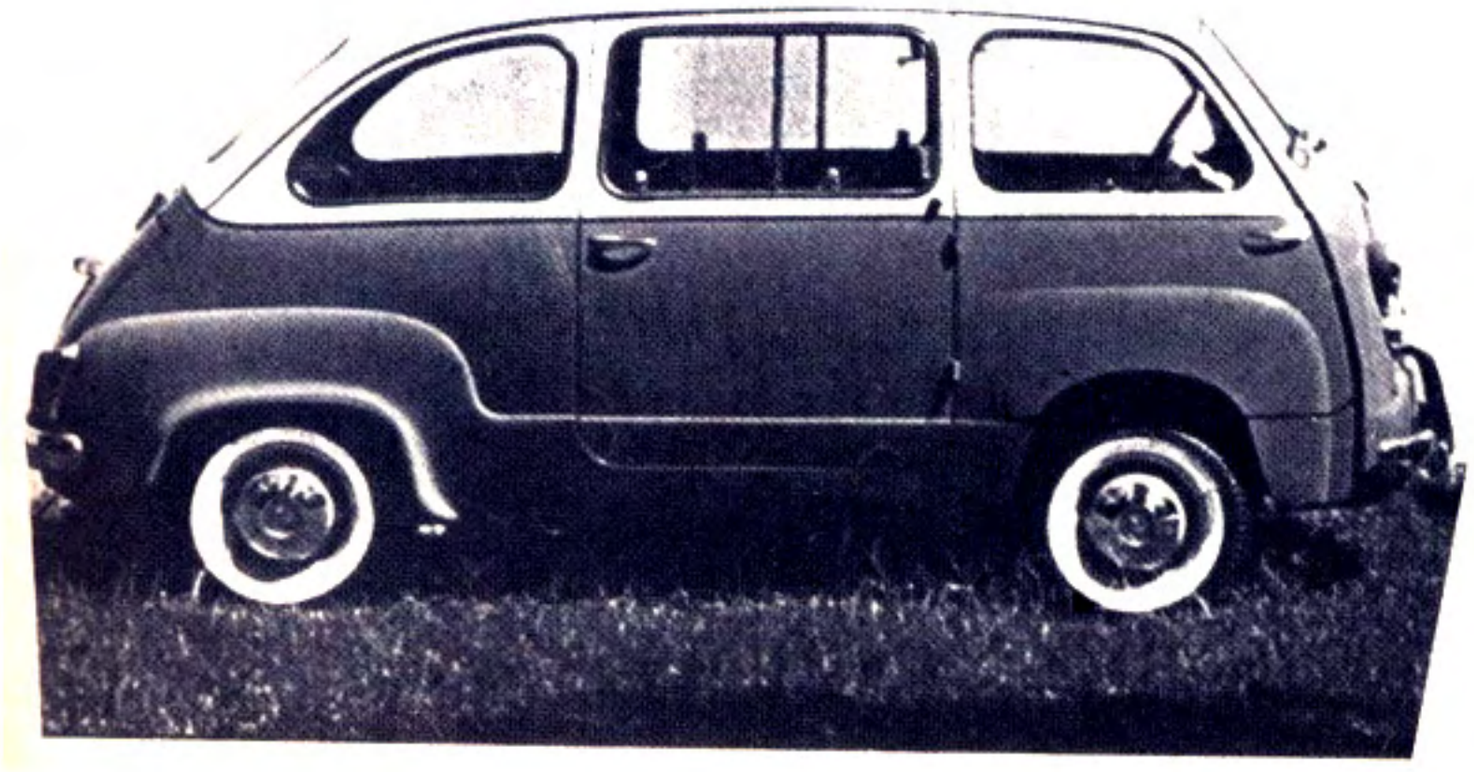
Dopo alcune schermate di presentazione viene proposto l'elenco dei parametri di scelta e viene richiesto di specificare per ciascuno di essi la sua importanza assegnando un valore da 0 a 100. Se, ad esempio, introduciamo i seguenti valori:

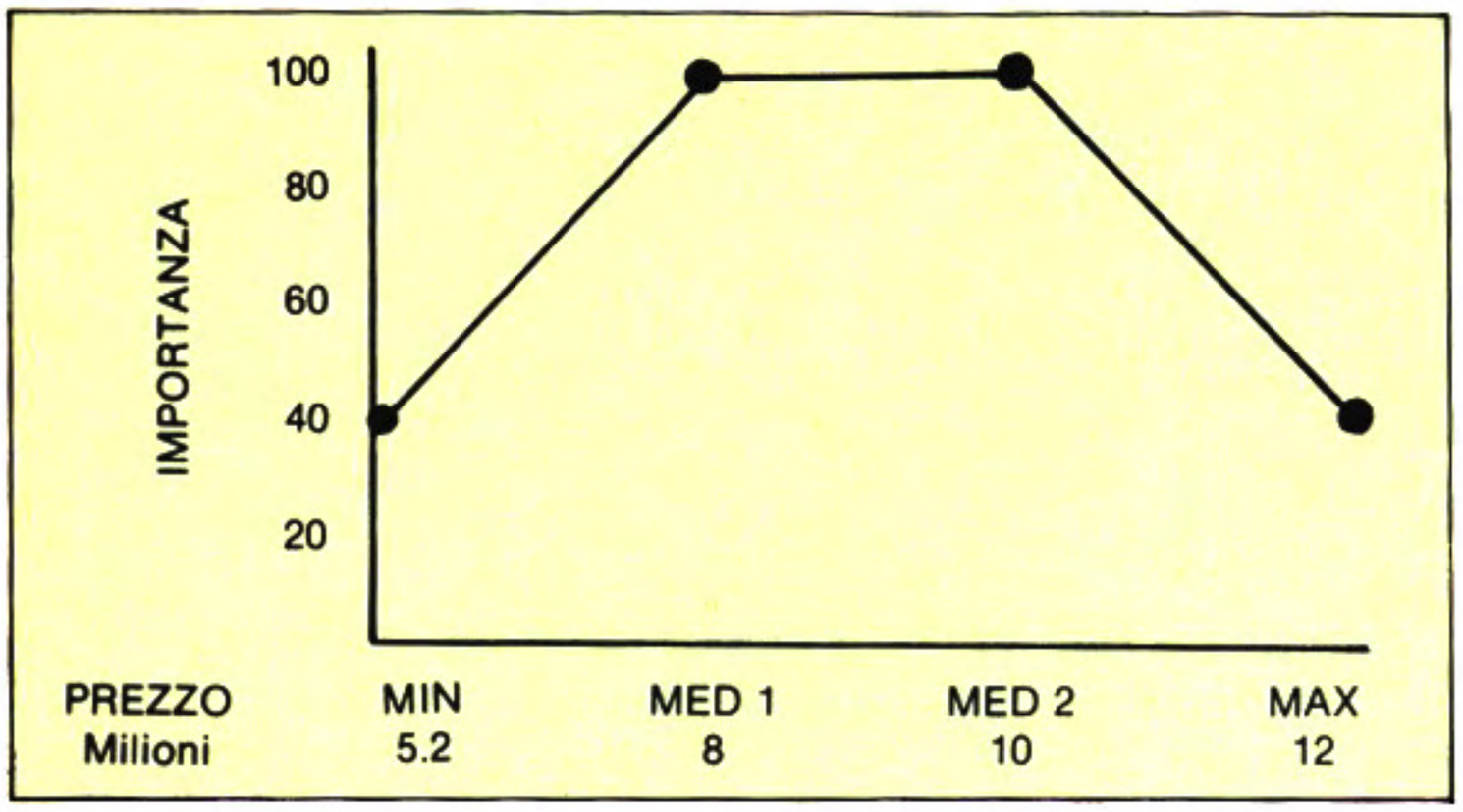
PREZZO	90
NUMERO PORTE	70
POTENZA	20
VELOCITA' MASSIMA	20
CONSUMI	100
PESO	0
DIESEL	0

Significa che sono per noi fondamentali il prezzo ed i consumi mentre non ci interessa né il peso della vettura né il fatto che si tratti di un motore Diesel oppure no. Poca importanza hanno poi la potenza e la velocità massima mentre è abbastanza importante il numero di porte.
Avremo più avanti la possibilità di variare l'importanza dei parametri di scelta in modo più dettagliato. Questa prima richiesta ha unicamente lo scopo di presentare i parametri tutti insieme per consentire la attribuzione di pesi relativi.
Dopo aver confermato i sette valori di importanza il programma inizia a presentare, uno alla volta, i parametri di scelta in forma grafica.
In verticale è riportata la importanza ed in orizzontale i valori che il parametro assume per le autovetture considerate.
Il primo parametro è il PREZZO e viene presentato il seguente grafico
I valori 5.2 e 17.4 sono rispettivamente il prezzo più basso



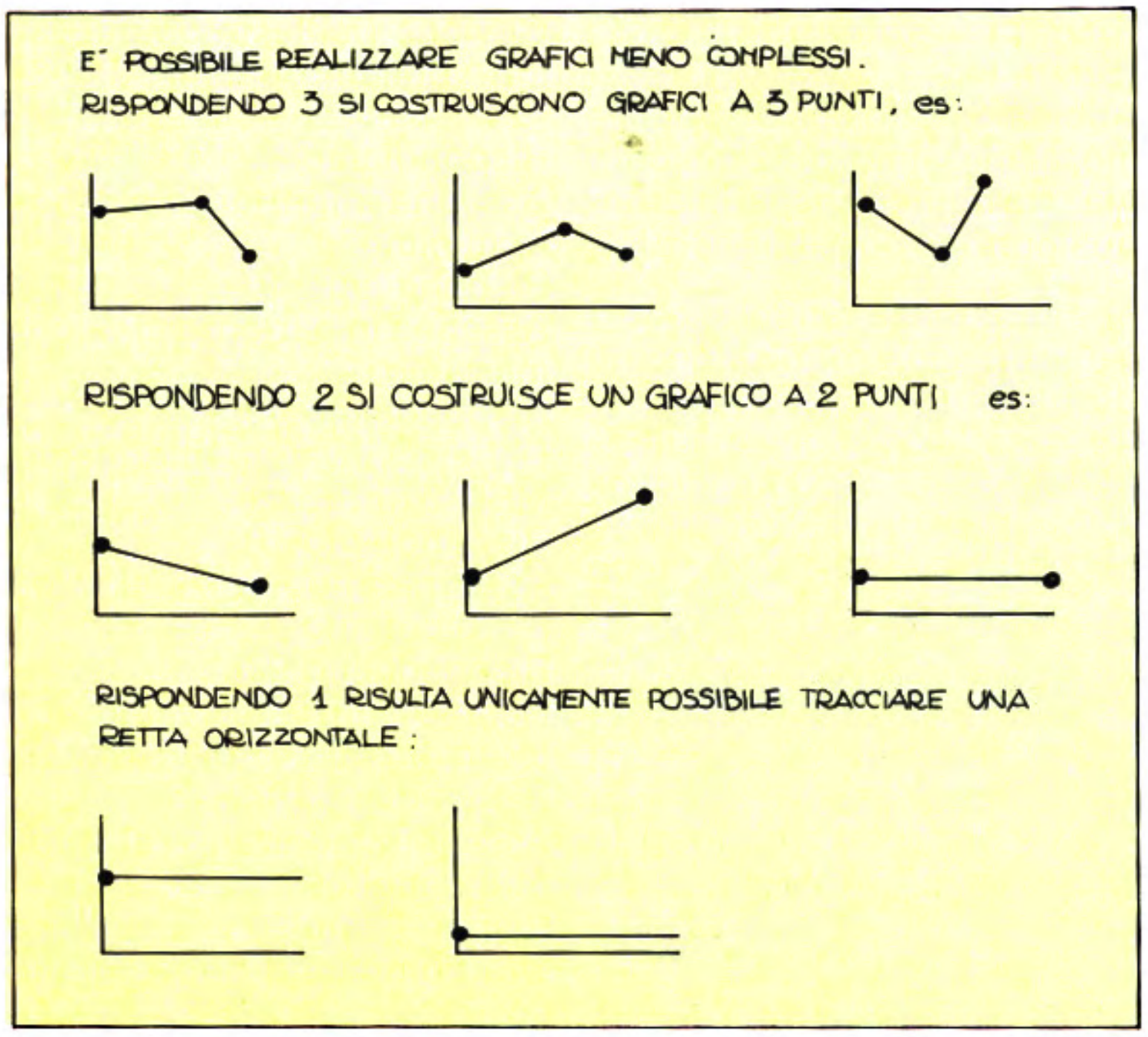
ed il prezzo più alto delle auto definite nel programma.
Il grafico mostra una retta in corrispondenza del valore 90 precedentemente assegnato.
Ci viene richiesto: OK? (S/N)
Se rispondiamo S viene assegnata importanza 90 a qualsiasi prezzo compreso fra i due valori mino e massimo. In tal modo le autovetture risulterebbero non differenziate.
Rispondiamo quindi N!
Ci viene offerta la possibilità di limitare il campo di variabilità del parametro. Se accettiamo di farlo rispondiamo S e ci viene richiesto di introdurre i nuovi valori minimo e massimo.
Se, ad esempio, disponiamo, al massimo, di 12 milioni per l'acquisto dell'auto potremo confermare 5.2 come valore minimo ed introdurre 12 quale valore massimo.
Risulteranno così automaticamente scartate dalla scelta tutte le autovetture di prezzo superiore ai 12 milioni. Il grafico viene aggiornato e ci ritorna la domanda: OK? (S/N).
Abbiamo detto che disponiamo, al massimo, di 12 milioni, ma vorremmo risparmiarne qualcuno se possibile. Ci piacerebbe spenderne da 8 a 10.
Vediamo come si può dire tutto ciò al nostro programma.
Rispondiamo N.
Alla richiesta se vogliamo limitare la variabilità risponderemo nuovamente N poichè abbiamo già operato una limitazione.
Ora ci viene richiesto se vogliamo variare l'importanza.
Rispondiamo S.
Ci viene richiesto quanti valori vogliamo definire 1-2-3-4?
Rispondiamo 4.
Assegna importanza al valore minimo.
Rispondiamo 40
Assegna in partenza al valore massimo.
Rispondiamo 40
Definisci il valore intermedio MED 2 di Prezzo.
Rispondiamo 10
Assegna importanza al valore 10 definito.
Rispondiamo 90
Definisci un altro valore intermedio MED 1 di PREZZO.
Rispondiamo 8
Assegna importanza al valore 8 definito.
Rispondiamo 90
Ora compare il nuovo grafico:





Abbiamo così comunicato al programma di assegnare importanza 90 alle vetture il cui prezzo cade nel campo che a noi interessa mentre per le vetture il cui prezzo è superiore ai 10 milioni o inferiore agli 8 milioni verranno assegnati importanze inferiori.
Potremo ora rispondere S alla domanda OK? (S/N) e ci apparirà il grafico del parametro successivo.

Come abbiamo visto, rispondendo 4 alla domanda QUANTI VALORI VUOI ASSEGNARE?, siamo giunti alla costruzione di un grafico costituito da 4 punti.



Si procede pertanto alla definizione dei grafici di importanza, uno per ciascun parametro di scelta.
Al termine il programma esamina, sulla scorta delle indicazioni ricevute, tutte le autovetture che conosce e le ordina secondo importanze decrescenti.
È questa una operazione che richiede qualche decina di secondi. Al termine appare il menù:

1 Per elencare le auto
2 Per rivedere le importanze
3 Per finire.

Rispondendo 1 vengono elencate le auto a partire da quella più vicina alle esigenze da noi specificate.
Rispondendo 2 vengono ripresentati i grafici dei parametri di scelta così come erano stati precedentemente definiti ed è possibile operare su di essi qualsiasi modifica.
È possibile aggiungere le descrizioni di altre auto e/o sostituire con altre auto quelle già definite.
Fate attenzione ad elencare per ciascuna vettura: ÷ *E, Progressivo, Descrizione* max 10 caratteri, *prezzo*, numero porte, *potenza, velocità massima, consumo medio, peso, diesel.*
In totale in ciascuna definizione di auto debbono comparire 9, virgole non una di più ne una di meno.
Buon divertimento!

CLAUDIO CAPPA

Questo mese utilizziamo lo spazio riservato alla posta dei lettori, per rispondere a tutti coloro che ci scrivono o ci telefonano lamentando problemi nel caricamento delle nostre cassette; se addirittura non ci rimandano subito la cassetta, dichiarandola diffettosa e pregandoci di sostituirla.
Quando però noi verifichiamo la cassetta, il risultato che ci attende è quasi sempre lo stesso: perfettamente funzionante.
Per evitare il prolungarsi di questa situazione, proviamo ad analizzare quali sono i problemi più frequenti nel caricamento dei programmi.
Nel gruppo di unità periferiche che attorniano il vostro TI 99, il registratore a cassette è senza dubbio la meno affidabile, essendo composto in gran parte di elementi meccanici ed essendo quasi sempre di tipo economico.
In caso di difficoltà nelle operazioni di «OLD» e «SAVE» quindi, il registratore è sempre il primo da sospettare.
La perfetta taratura di un registratore a cassette, che nell'uso analogico ha una importanza relativa, diventa invece basilare quando lo usiamo per registrare e riprodurre segnali digitali.
Le grandezze che entrano in gioco sono: il volume in riproduzione, la velocità di scorrimento del nastro e l'allineamento delle testine.
Vi sono poi le operazioni di manutenzione periodica che, se tralasciate, vanno ad influenzare la risposta in frequenza e quindi la forma del segnale digitale riprodotto.
Vediamo quindi nel dettaglio i problemi che possono affliggere il vostro registratore a cassette, nell'ordine di probabilità con cui si presentano, e le operazioni di controllo e taratura da effettuare per eliminarli.
— Il computer non accetta un qualsiasi livello di segnale dal registratore; è quindi necessario verificare sperimentalmente il volume più adatto per caricare i programmi; inoltre, cassette di provenienza diversa, possono essere incise a livelli diversi, per cui, se ad un primo tentativo il programma non viene caricato, conviene riprovare cambiando volume.
In genere, l'errore «NO DATA FOUND», significa che il volume è troppo basso, «ERROR DETECTED IN DATA», significa che è troppo alto.
Durante l'operazione di «SAVE», il problema non si pone poiché, in genere, questi apparecchi hanno il livello di registrazione automatico.
— Le testine e il rullino di trascinamento del nastro (pinch roller), si sporcano a causa dei residui lasciati dal nastro magnetico; è necessario provvedere alla loro pulizia almeno ogni 30 ore di utilizzo.
Usate semplicemente bastoncini del tipo «cotton fiocc» imbevuti di alcool.
— Oltre allo sporco, le testine accumulano una magnetizzazione residua che influisce negativamente sulla riproduzione. Per eliminarla potete usare, sempre ogni 30 ore, uno smagnetizzatore per testine, oppure, una cassetta elettronica «Head demagnatizer», del costo di circa 30.000 lire, che si inserisce nel registratore come una normale cassetta.
— La velocità di scorrimento del nastro nei registratori a cassette è standardizzata a 4.75 cm/s; se il vostro registratore, probabilmente per vecchiaia, altera la velocità oltre i limiti di tolleranza, varierà di conseguenza il numero di Baud al secondo dei dati che affluiscono al computer e il vostro TI 99 non li accetterà.
Per controllare la velocità, se non possedete un oscilloscopio potete usare un'altra cassetta speciale: la cassetta «strobo», che utilizza il leggero effetto stroboscopico di una qualsiasi lampadina.
— L'ultimo elemento da controllare è l'azimuth, ovvero l'allineamento della testina di registrazione/riproduzione.
Sopra la testina, esiste nel registratore un foro per accedere ad una vite che regola la posizione della testina rispetto al nastro; questa deve essere regolata per avere il massimo segnale in uscita.
Per questa taratura è sufficiente una cassetta con un segnale a livello costante (solitamente la cassetta strobo porta sul nastro una nota a livello costante), ed un qualsiasi strumento che indichi il livello di uscita del registratore, al limite, anche il vostro orecchio.
Se il registratore avrà superato tutte queste prove, sarà ben difficile che il vostro TI 99 non accetti un programma, a meno che non sia stato registrato da un amico con il registratore starato!

TI 99 NEWSOFT

L. 10.000

Numero 10 - anno II° - giugno 1985

COME SI COSTRUISCE UN ADVENTURE

I COCKTAILS ESTIVI CON IL TI 99

IN SOFTWARE & SOFTWARE
L'IMPICCATO - DISCOVERY
GOLF

SOMMARIO

☐ **TI BASIC**
Come risparmiare la memoria del vostro computer pag. 2

☐ **RECENSIONI**
Questo mese parliamo della cassetta ''Family Game Pack'' pag. 3

☐ **TI EXTENDED**
La grafica multisprites pag. 4

☐ **TI 99 4/A**
I dispositivi di input e output pag. 6

☐ **L'IMPICCATO**
Un classico che non poteva mancare pag. 9

☐ **EXTENDED + 32 K**
Il multicolor mode pag. 10

☐ **SCROL DEL VIDEO**
Una routine in linguaggio macchina per i vostri giochi pag. 11

☐ **PROGRAMMAZIONE**
Vi spieghiamo come si costruisce un adventure pag. 12

☐ **DISCOVERY**
Una avventura nel Far-West pag. 14

☐ **TRE DIMENSIONI**
Solidi in prospettiva pag. 16

☐ **BIT MAP MODE**
La gestione del modo grafico in Bit Map pag. 18

☐ **GOLF**
Un minigolf elettronico pag. 20

☐ **EASY COCKTAILS**
Il programma di copertina di questo mese pag. 22

☐ **INVENZIONI**
Apparecchiature di gioco alla TV pag. 25

☐ **PICCOLI ANNUNCI** pag. 26

☐ **LA VIGNETTA** pag. 29

☐ **COLLABORATE CON NOI** pag. 31

☐ **COMPUMARKET** pag. 33

TI 99 NEWSOFT
ANNO II° - Numero DIECI
GIUGNO 1985
MENSILE
Direttore responsabile:
ANTONIO C. LOSITO
Coordinatore Editoriale:
VINICIO CIVININI
Direttore Amministrativo:
ENRICO CALLERIO
Art Director:
BEPPE RE-FRASCHINI
Illustrazioni:
MARIA TERESA COCOZZA
Segreteria:
CRISTINA FRISON
Redazione:
FIORENZA AURIEMMA
Collaboratori:
SERGIO BORSANI - GIANPAOLO BOTTIN - DANIELE CATALFAMO - PAOLO CIVARDI - STEFANO DELLI PONTI - FRANCESCO MACALI - MAURO MATTIAZZI - EZIO MONTINI - LEVIO PEROTTI - RICCARDO ROSSI - MARCO SQUINTANI - ALBERTO STRAFILE - SEBASTIANO TOMASELLO
TI 99 NEWSOFT è una pubblicazione della **NEWSOFT snc.**
Via Stefano Jacini, 4 - 20121 MI
Tel. 02/807464 - Telex 324284
REGISTRAZIONE AL TRIBUNALE DI MILANO N. 323 - DEL 7.7.84

© COPYRIGHT NEWSOFT snc
TUTTI I DIRITTI RISERVATI
Manoscritti e foto originali, anche se non pubblicati, non si restituiscono ed è vietata la riproduzione anche parziale di testi, fotografie e listati.

Pubblicità: AV STUDIO
Via Giuseppe Meda, 23
Tel. 02/8357155
Abbonamenti: l'abbonamento annuo a 11 numeri di **TI 99 NEWSOFT** costa L. 95.000 per l'Italia; L. 160.000 per l'Europa (spedizione aerea); L. 200.000 per tutti gli altri paesi. L'importo, per assegno non trasferibile o per vaglia postale, va inviato a **NEWSOFT snc,** Via S. Jacini, 4 - 20121 Milano
Stampa:
lalitotipo s.r.l. milano
Concessionaria per la distrib.:
EUROSTAMPA Srl - C.so Vittorio Emanuele, 11 - 10123 Torino
Telefono 011/538166-7

L'editore non si assume responsabilità alcuna sul contenuto degli articoli e/o software inviati.
Per gli articoli firmati e/o siglati da collaboratori esterni, la redazione si assume la responsabilità prevista dalle leggi sulla stampa.

Tutti possono collaborare a TI 99 NEWSOFT, sia con articoli che con programmi. Scriveteci o telefonateci in redazione per avere tutti i ragguagli.

RISPARMIARE MEMORIA

La memoria disponibile all'utente del nostro computer (per chi l'avesse scordato) è di 16K bytes, parecchi per un personal di questa categoria. Aumentarla, per chi ha l'espansione da 32K, è cosa facile; ma, per chi non ha questa possibilità, l'impresa si rivela piuttosto ardua. Tuttavia, basta sostituire la parola aumento con il termine risparmio per risolvere almeno qualche piccolo problema.

Esistono infatti dei sistemi di controllo della memoria utilizzata e utilizzabile con approssimazioni in eccesso o per difetto. Uno dei più conosciuti è fare 1x = x + 9 2 GOSUB 1; dopo il RUN attendere il MEMORY FULL e digitare PRINT 16-X per avere la memoria rimasta. Bisogna però avere l'accortezza di eliminare queste due linee (da porsi all'inizio del programma) prima di mandare in esecuzione lo stesso o di salvarlo su nastro (magari per mandarlo alla nostra redazione).

Vediamo ora alcuni consigli pratici per evitare che il programma si blocchi per la mancanza di bytes liberi. Si ha già un bel risparmio di memoria (e di velocità) eliminando tutte le REM poichè ogni statement di questo tipo utilizza almeno un byte, oltre a quelli che ne compongono il testo o il messaggio. Usate preferibilmente le variabili invece delle costanti in casi del genere: se dovete utilizzare il valore numerico costate di 3.141592654 (il pi greco che esiste come istruzione in EXTENDED) più volte nello stesso programma, chiamatelo — per esempio — P. Otterrere già ottimi risultati.
Sempre sulle variabili: chiamate, se possibile, con il medesimo nome variabili diverse. Altro esempio: la K da una CALL KEY (O,K,S) può essere tranquillamente adoperata per definire una FOR K = 1 TO 100 di qualche riga più avanti. E il discorso vale anche per le variabili di stringa, tipo A$, sfruttabili quando si deve rispondere a domande diverse in momenti diversi ma con la stessa procedura di risposta. Es.: 10 INPUT ''CONTINUO (S/N)'': A$.

Evitate di mettere i comandi END e STOP alla fine fisica di un programma. Sfruttate l'istruzione GOSUB per eseguire porzioni di programma che compiono azioni uguali.

Oltre a questi piccoli accorgimenti, ci sono poi dei veri e propri trucchi da programmatore ''navigato'', come usare: per visualizzare una riga vuota o sfruttare più di quattro linee di testo (V. TI 99 New Soft n. 8).

Se infine non ricordate più questi utili suggerimenti, allora l'unico rimedio per risparmiare la memoria del computer è quello di migliorare la vostra con una buona cura di fosforo...

MARCO SQUINTANI

FAMILY GAME PACK

Programma:
«Family Game Pack»
della Pewterware.

Configurazione richiesta:
consolle + registratore + cavo.

IL GIOCO

Sotto un unico titolo si celano tre diversi giochi di abilità, uno di carte (Challenger Poker), uno di dadi (Roll Five) e un terzo di memoria (Match Wits). Vediamoli in dettaglio.

Il primo programma registrato sulla cassetta è Challenger Poker. Una volta mandato in esecuzione, il gioco mostra una scritta, ***Startup in Progress,*** che significa inizializzazione in atto o partita; la stessa dicitura viene ripresa anche nei giochi successivi. In ogni caso, in tutti e tre, vi possono prender parte fino a 4 giocatori, che verranno contraddistinti, nel caso di Challenger Poker, con le iniziali del proprio nome e cognome (due lettere, quindi). Superfluo dire che occorre raggiungere il miglior punteggio possibile, realizzando sul tavolo alcune combinazioni vincenti.

Si possono sistemare in modo opportuno le carte con comandi da tastiera, come da istruzioni allegate. Vincerà chi per primo riuscirà a totalizzare 100 punti o realizzerà il punteggio più alto.

Di seguito è registrato ***Roll Five*** che, rispetto al precedente, presenta anche una piacevole, seppur breve, musichetta di sottofondo. Qui potrete inserire tre lettere per distinguervi dagli avversari durante la partita.

Dopo un attimo di pausa appariranno i dadi con le varie combinazioni possibili, che vanno dal full (una coppia e un tris) alla scala (straight) al poker, ecc. ecc. Anche in questo caso, comunque, potrete avere maggiori ragguagli consultando il foglietto illustrativo.

Il gioco ha termine dopo il trentesimo turno dell'ultimo giocatore e vincerà, naturalmente, chi avrà fatto il punteggio più alto.

Per concludere, ***Match Wits*** (che ricorda molto l'*Hidden Pair* della Texas Instruments). Bisogna infatti trovare il simbolo che fa coppia con quello indicato al centro, nel corso del vostro turno, col mucchio di piastrine blu, gialle, verdi o rosse - a seconda del giocatore - che sono coperte. Il giro è in senso orario e finisce allorquando una delle persone in lizza, avrà trovato tutte e 12 le coppie di simboli uguali.

La grafica è ben curata e dà effetti suggestivi anche nell'intestazione. Non avrete sicuramente modo di annoiarvi, soprattutto se sarete in buona compagnia.

MARCO SQUINTANI

GRAFICA MULTISPRITES

Anche in questa occasione vi proponiamo ulteriori approfondimenti sull'argomento SPRITES. Ci soffermiamo così a lungo su questo determinato argomento, perchè riteniamo sia uno tra i più importanti, nell'ambito della programmazione di giochi.
Senza alcun dubbio la gestione degli sprites in extended non è semplice quanto si possa immaginare, infatti il TI 99, come d'altra parte tutti i computer, ha i suoi difetti, ma con pazienza e volontà tutto è risolvibile e ci pare di averlo già dimostrato in qualche occasione.
Questo mese tratteremo una tecnica di animazione e al tempo stesso di effetti stupefacenti.
Vi faremo vedere come sfruttare un piccolo ''BUG'' della gestione degli sprites, per ottenere effetti ottici di movimento.
Questa tecnica da noi soprannominata MULTISPRITES, è molto usata dai programmatori americani.
Essa consiste nel sovrapporre più di uno sprites per ottenere una sola figura, permettendo di utilizzare più di un colore per ogni sprites.
Per più di un colore, non si intende che miracolosamente il TI 99 ci permetta di definire sprites ''multicolor!'', bensì, intendiamo dire che se un ''omino'' deve avere come esempio i pantaloni verdi, possiamo farli, creando uno sprite di colore verde che abbia la forma dei pantaloni desiderati, ecc...
Per poter ottenere tale effetto dobbiamo sottostare a determinate regole. Dette regole si basano sul fatto che gli sprites hanno una loro precedenza di visualizzazione.
Ad esempio se noi sovrapponiamo lo sprite numero * 1 al * 2, otterremo, che lo sprite con valore inferiore avrà precedenza, nella visualizzazione, su quello con numero superiore.
Il risultato di tale esperimento sarà che: solo lo sprite * 1 sarà visualizzato, mentre il * 2 sarà cancellato dallo schermo sino a quando * 1 non si sarà spostato dalla sua posizione.
Il caso che ci interessa è esattamente l'opposto, infatti se noi proviamo a sovrapporre due sprites con valori * 1 sotto e * 2 sopra, vedremo che nessuno dei due sprites viene cancellato dal video e le figure sono sovrapposte.
Questa sovrapposizione ci permette di attuare la tecnica della grafica multisprites che sfrutta proprio questa particolare caratteristica.
Per chiarirvi meglio le idee in proposito, abbiamo pensato di fornirvi un programma, che troverete nella cassetta, che usa detta tecnica.
Il programma vi fornirà, oltre che un utile esempio, anche un piacevole passatempo. Il progamma in questione è: ZAXXON creato dai nostro programmatori appositamente per favorire l'apprendimento.

Zaxxon si può definire un gioco multifunzione.
Il gioco è strutturato in quattro parti principali:

1) Routine di caricamento caratteri e posizionamento degli sprites.

2) Routine di movimento avversario ed eroe.

3) Routine di verifica collisione tra sprites.

4) Routine di inserimento dati menu principale e visualizzazione responso di fine partita.

Una volta mandato in esecuzione, il programma ci proporrà quattro tipi di gioco:
1) Attacco;
2) Fuga;
3) Attacco con bombe a contatto;
4) Fine del gioco.

Nella prima fase dovremo colpire e distruggere le postazioni radar e i tanker di carburante.
Nella seconda dovremo fuggire dai colpi lanciati dai nemici.
Nel terzo per poter distruggere le basi nemiche, dovremo passarci sopra.
Una volta operata la scelta di gioco, ci verrà richiesto un tempo di gioco, esso varia da 0 a 999 a seconda della nostra volontà, basta digitare il numero di tempo voluto e quindi premere enter.
Il gioco si presta molto ad eventuali modifiche.
Non ci resta quindi che augurarvi buon divertimento.

EZIO MONTINI

```
100 !-----------------------
110 ! PROGRAMMA CREATORE DEI
120 !  SOTTOPROGRAMMI IN LM
130 !  SDOWN    &    SLEFT
140 !          BY
150 !     MONTINI EZIO
160 !-----------------------
170 CALL INIT
180 CALL LOAD(16368,83,76,69,70,84,32,40,70)
190 CALL LOAD(16376,83,68,79,87,78,32,40,24)
200 CALL LOAD(8194,40,144,63,240)
210 CALL LOAD(10032,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,
128,128,128,128,128,128,128)
220 CALL LOAD(10054,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,
128,227,227,227,227,227,227)
230 CALL LOAD(10076,227,227,227,227,227,227,227,227,227,227,227,227,227,227,227,
227,227,227,227,227,227,227)
240 CALL LOAD(10098,128,128,128,128,164,165,166,128,174,207,142,128,159,128,136,
145,128,141,128,146,128,137)
250 CALL LOAD(10120,128,128,145,128,128,128,128,128,128,128,128,128,128,128,128,227,
227,227,227,227,227,227,227)
260 CALL LOAD(10142,227,227,227,227,227,227,227,227,227,227,227,227,227,227,227,227,
227,227,227,227,227,128,128)
270 CALL LOAD(10164,128,128,176,210,197,211,211,128,165,174,180,165,178,128,215,
200,197,206,128,198,201,206)
280 CALL LOAD(10186,201,211,200,197,196,128,128,128,128,128,0,255,0,255,0,255,0,
255,0,255,0,255)
290 CALL LOAD(10208,0,255,0,255,0,255,0,239,0,255,0,255,0,255,0,239,0,255,0,255,
128,128)
300 CALL LOAD(10230,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,
128,128,128,128,128,128,128)
310 CALL LOAD(10252,128,128,128,128,128,128,128,128,128,128,128,0,0,194,139,2,0,0,0,
2,1,36,244)
320 CALL LOAD(10274,2,2,2,224,4,32,32,44,2,0,0,32,4,32,32,36,2,0,0,0,2,1)
330 CALL LOAD(10296,39,244,2,2,0,32,4,32,32,36,194,202,4,91,194,139,2,0,0,1,2,1)
340 CALL LOAD(10318,36,244,2,2,0,31,4,32,32,44,6,0,2,1,36,244,2,2,0,31,4,32)
350 CALL LOAD(10340,32,36,2,32,0,32,2,128,2,255,26,238,6,160,40,120,194,202,4,91
,2,0)
360 CALL LOAD(10362,0,31,2,1,128,0,4,32,32,32,2,32,0,32,2,128,2,255,26,247,4,91)
370 CALL LOAD(10384,0,255)
```

```
 99/4 ASSEMBLER                                                  PAGE 0001
VERSION 1.2
  0001                    DEF SDOWN,SLEFT
  0002
  0003 0000         BUF0  BSS >02FF
  0004
  0005 0300 8080    BUF128 DATA >8080,>8080,>8080,>8080,>8080
       0302 8080
       0304 8080
       0306 8080
       0308 8080
  0006 030A 8080           DATA >8080,>8080,>8080,>8080,>8080
       030C 8080
       030E 8080
       0310 8080
       0312 8080
  0007 0314 8080           DATA >8080,>8080,>8080,>8080,>8080
       0316 8080
       0318 8080
       031A 8080
       031C 8080
  0008 031E 8080           DATA >8080,>8080,>0000
       0320 8080
       0322 0000
```

```
0009
0010
0011
0012 0324 C28B   SDOWN   MOV   R11,R10
0013 0326 0200           LI    R0,>0000
     0328 0000
0014 032A 0201           LI    R1,BUF0
     032C 0000'
0015 032E 0202           LI    R2,>02E0
     0330 02E0
0016 0332 0420           BLWP  @>202C
     0334 202C
0017 0336 0200           LI    R0,>0020
     0338 0020
0018 033A 0420           BLWP  @>2024
     033C 2024
0019 033E 0200           LI    R0,>0000
     0340 0000
0020 0342 0201           LI    R1,BUF128
     0344 0300'
0021 0346 0202           LI    R2,>0020
     0348 0020
0022 034A 0420           BLWP  @>2024
     034C 2024
0023 034E C2CA           MOV   R10,R11
0024 0350 045B           B     *R11
0025
0026
0027 0352 C28B   SLEFT   MOV   R11,R10
0028 0354 0200           LI    R0,>0001
     0356 0001
0029 0358 0201   CONT    LI    R1,BUF0
     035A 0000'
0030 035C 0202           LI    R2,>001F
     035E 001F
0031 0360 0420           BLWP  @>202C
     0362 202C
```

```
 99/4 ASSEMBLER
VERSION 1.2
  0032 0364 0600           DEC   R0
  0033 0366 0201           LI    R1,BUF0
       0368 0000'
  0034 036A 0202           LI    R2,>001F
       036C 001F
  0035 036E 0420           BLWP  @>2024
       0370 2024
  0036 0372 0220           AI    R0,>0020
       0374 0020
  0037 0376 0280           CI    R0,>02FF
       0378 02FF
  0038 037A 1AEE           JL    CONT
  0039 037C 06A0           BL    @CANC
       037E 0384'
  0040 0380 C2CA           MOV   R10,R11
  0041 0382 045B           B     *R11
  0042
  0043 0384 0200   CANC    LI    R0,>001F
       0386 001F
  0044 0388 0201   CT      LI    R1,>8000
       038A 8000
  0045 038C 0420           BLWP  @>2020
       038E 2020
  0046 0390 0220           AI    R0,>0020
       0392 0020
  0047 0394 0280           CI    R0,>02FF
       0396 02FF
  0048 0398 1AF7           JL    CT
  0049 039A 045B           B     *R11
  0050
  0051                     END
  0000 ERRORS
```

DISPOSITIVI DI INPUT/OUTPUT

LA TASTIERA

Entriamo nella parte più prettamente tecnica dell'argomento, dopo aver trattato la volta scorsa la tastiera del TI 99 da un punto di vista generale.

La nostra Keyboard è collegata al bus dati del computer tramite una serie di quindici fili multicolori e uniti da una guida trasparente (nella versione per l'Italia).

I tasti sono invece saldati su un'unica piastra-madre di colore verde chiaro che, alla luce del sole, mette in evidenza le varie connessioni e piste circuitali.

Sulla carta risulta essere messa in comunicazione con il coprocessore di input/output (I/O) TMS 9901 che si collega poi, a sua volta, con la CPU attraverso il bus A10-14 e quindi al TMS 9918 VDP, che visualizza il messaggio sullo schermo (figura 1).

Abbiamo, comunque, più modi di gestire la tastiera: modo normale, shiftato, modo control e modo function. Vediamoli ora in dettaglio.

Il *modo normale* si ha con il tasto Alpha Lock alzato o libero e si ottengono caratteri minuscoli al completo, numeri, punteggiatura ridotta a punto, virgola, punto e virgola, tratto obliquo e simbolo di eguale (figura 2).

Con l'*alpha lock* abbassato o bloccato si ha lo stesso effetto, ma con il set di caratteri alfabetici in maiuscolo. Questa è la versione consigliata per una miglior impostazione e utilizzo dei programmi.

Modo Shiftato: si ottiene premendo contemporaneamente al tasto desiderato quello di shift, posto all'estrema sinistra e destra della seconda riga a partire dal basso. Vengono visualizzati caratteri alfabetici maiuscoli, punto esclamativo,

simboli di at, cancelletto, simbolo del dollaro, del per cento, dell'innalzamento a potenza, della e commerciale (&), dell'asterisco o simboli di moltiplicazione, parentesi tonda aperta e chiusa al posto dei caratteri numerici, trattino breve o simbolo di sottrazione, due punti, simbolo di maggiore e minore al posto di altri (figura 3). Questo modo, come gli altri che seguono, non è permanente; cioè, una volta rilasciato il tasto di cambio modo, tutto torna ''normale''.

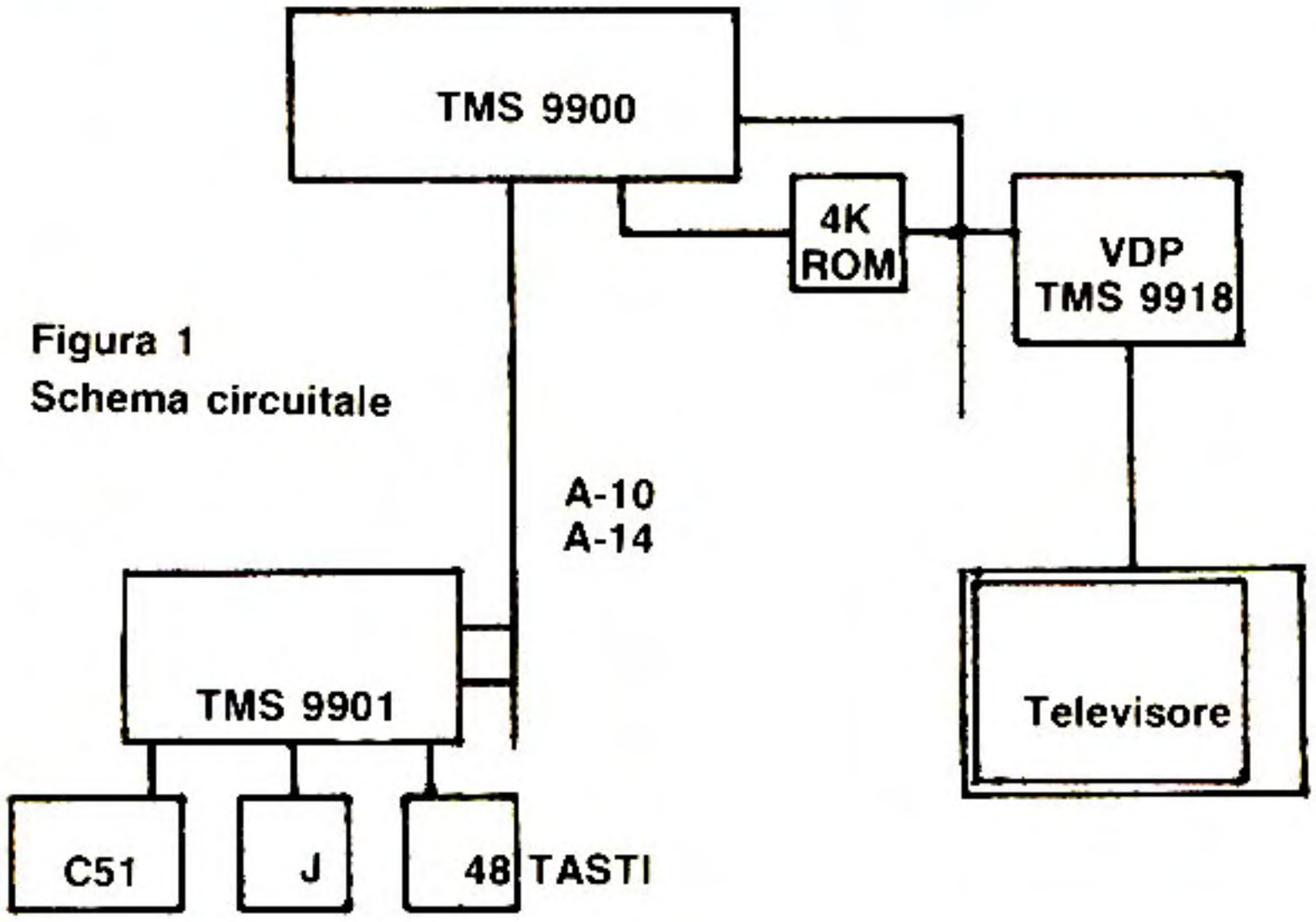


Figura 1
Schema circuitale

Interfaccia Joystick Tastiera

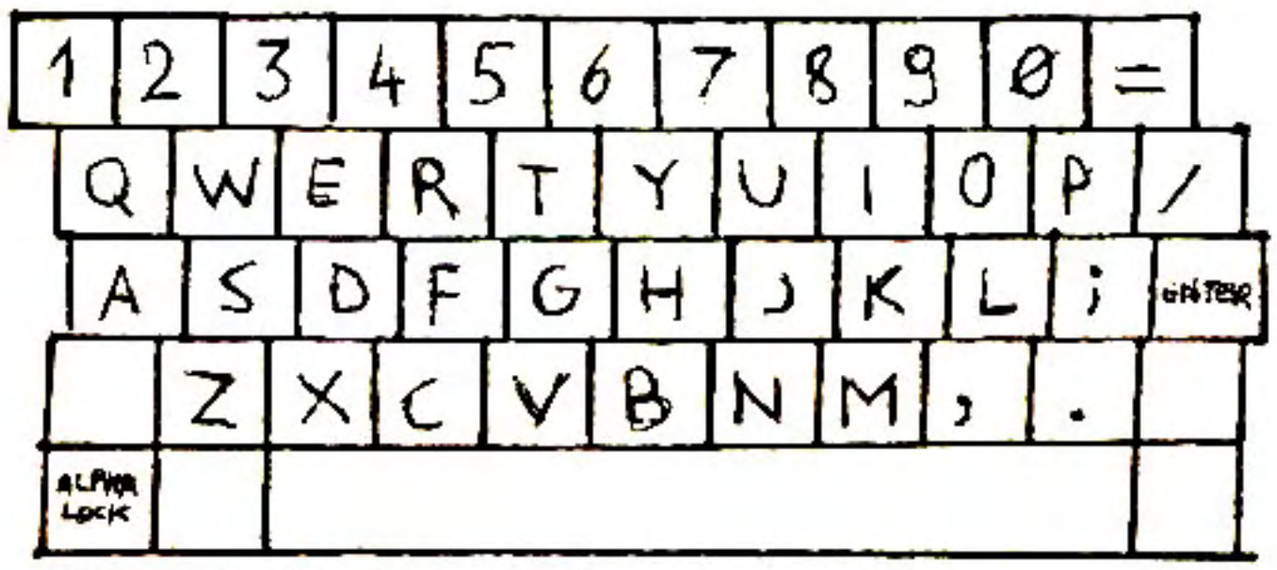


Figura 2 - Modo normale

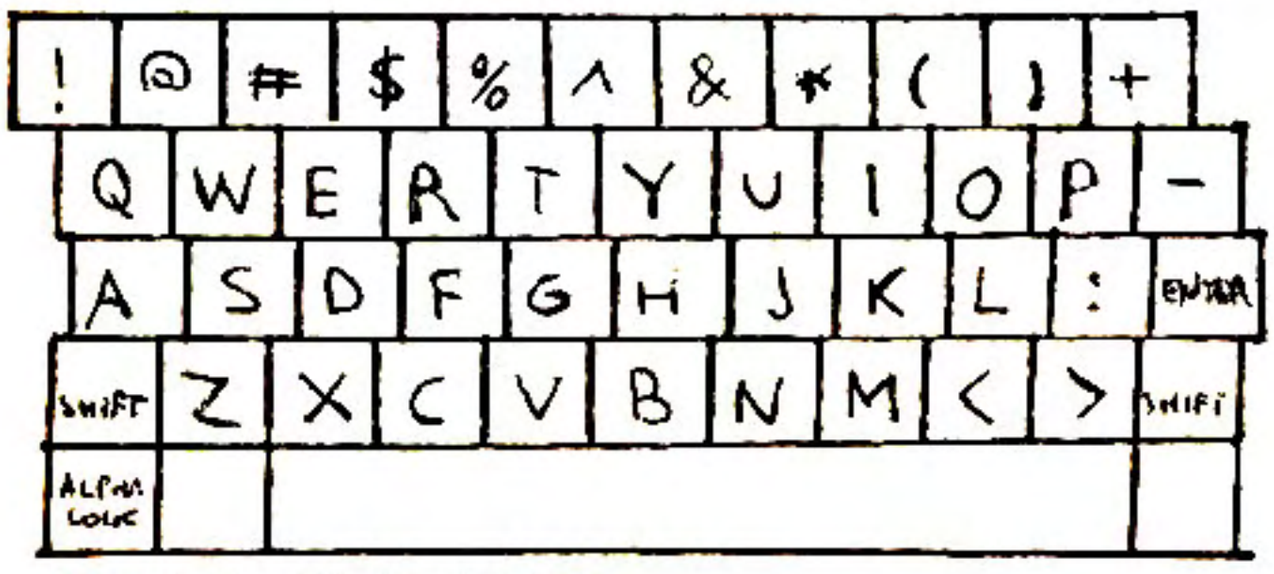


Figura 3 - Modo shiftato

Modo Control: in questa particolare situazione tutti i caratteri della tastiera assumono un diverso valore di codice ASCII; esattamente, per il TI Basic, dal 127 al 159 apparendo con la definizione grafica in memoria in quel momento. Quando si accende il computer, o dopo aver dato un comando di reset (BYE, QUIT), tutti i caratteri vengono visualizzati come lo spazio, pur conservando il loro codice ASCII. Provando infatti a battere una control qualsiasi (CTRL + tasto) e premendo ENTER si avrà il messaggio generico di errore BAD NAME, cosa che non avviene inserendo lo spazio.

L'utilità di queste control sta nel fatto che la loro definizione può essere controllata e voluta dall'utente in questa maniera: battete direttamente (senza numero di linea) un'istruzione di CALL CHAR ridefinendo - come nell'esempio di figura 4 - il carattere 129 e poi premete ENTER. Provate adesso a digitare control e il tasto A. Apparirà una serie di linee orizzontali alternate o, per meglio dire, il vostro personale carattere 129. Tale carattere resterà così immutato per il tempo che vorrete; finché, in altre parole, non darete un BYE o uscirete dal linguaggio (o lo ridefinirete in altro modo).

Per i più smaliziati è doveroso dire che il carattere 129 è utilizzato anche dal computer (e quindi modificato) mentre svolge operazioni di caricamento To salvataggio di dati e programmi su supporto magnetico. Sull'altra grande caratteri-

stica (poter ottenere le parole chiave del basic direttamente) dovreste sapere ormai tutto, specialmente se avete letto con attenzione la rubrica TI BASIC del numero scorso.

Modo Function: quest'ultima disposizione della tastiera risulta abilitata premendo il tasto FCTN contemporaneamente ad un altro. Ci pare giusto soffermarci un po' più a lungo e dettagliatamente sul singolo uso dei vari tasti che, d'ora in avanti, per semplicità, verranno intesi unitamente alla pressione del tasto FCTN.

Il tasto 1 (DELete) permette di cancellare, ogni volta che è premuto, il carattere immediatamente sotto a quello del cursore; il 2 (INSert) inserisce il carattere da voi scelto, a destra rispetto alla posizione del cursore, e si disinserisce premendo un altro carattere di funzione (FCTN D per esempio) o l'ENTER; il 3 (ERASE) cancella completamente tutto ciò che si trova nella linea occupata dal cursore, e lo riposiziona all'inizio della stessa; il tasto 4 (CLEAR) blocca l'esecuzione di un programma o il suo listaggio e annulla, pur lasciandolo visualizzato, un comando inserito erroneamente.

I tasti che vanno dall'1 allo 0 svolgono funzioni particolari in altri linguaggi, sul cui uso rimandiamo senz'altro all'esauriente documentazione dei relativi manuali.

Per quanto riguarda i tasti rimanenti, essi provocano lo stesso effetto delle control (carattere vuoto) o visualizzano i caratteri di tilde o accento circonflesso, parentesi quadra aperta e chiusa, tratto lungo e basso, punto interrogativo, accento o apostrofo, virgolette o apici, linea verticale discontinua, parentesi graffa aperta e chiusa, tratto greve obliquo, in più, naturalmente, abilitano i tasti di controllo del cursore così spesso utilizzati nei giochi per supplire alla mancanza del joystick (figura 5).

Fa spicco, al centro, la voluminosa e longilinea barra spazio, attorniata dai tasti contrassegnati da un pallino colorato rosso, giallo e bianco (control; enter e function nell'ordine). La parte del leone in una qualsiasi tastiera la fa, tuttavia il tasto di ENTER che, per altri, è il RETURN o C/R e che conferma al computer la nostra decisione.

Nell'affrontare l'argomento è stata posta la più grande cura nell'abbracciare ogni possibile aspetto della nostra keyboard, quasi rasentando la pignoleria e la prolissità. Il miglior risultato che possiamo auspicare resta comunque quello di avervi spiegato in modo chiaro e semplice il funzionamento di questo importante dispositivo. Premiate il nostro sforzo continuando a seguirci anche per i prossimi argomenti.

MARCO SQUINTANI

```
CALL CHAR (129, ''FFOOFFOOFFOOFF'')
```

Figura 4 - Esempio di definizione di una control

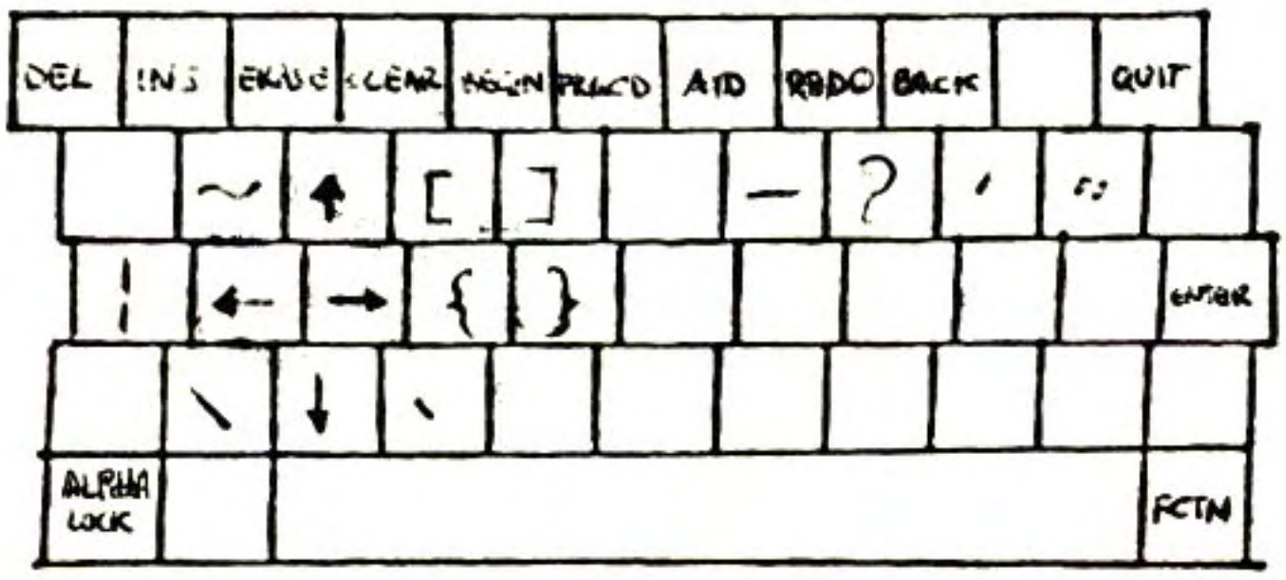


Figura 5 - Modo function

CONOSCETE IL GIOCO DELL'IMPICCATO?

Non dite di no! è un classico gioco, come battaglia navale, delle lezioni scolastiche più tediose, ultima arma di tanti studenti, un po' somari, recalcitranti ad imparare impossibili declinazioni latine.
Nella versione per computer, buona per i giorni piovosi, bisogna indovinare il nome di una città italiana di cui solo la lettera iniziale e quella finale sono svelate.
I trattini intermedi fanno parte della parola e costituiscono un aiuto all'identificazione del numero totale di lettere costituenti il nome della città.
Facile vero?
Forse non tanto; avete a disposizione 11 tentativi per indovinare; ogni risposta sbagliata farà apparire sullo schermo un dettaglio della procedura di impiccagione; ultimati i preparativi sarete bruscamente trattati alla maniera con cui gli abitanti del vecchio West processavano i ladri di cavalli.
E, per complicarvi un poco la vita, una vocale sbagliata sarà considerata come doppio errore.
Le province italiane, poi, sono 82 (lo sapevate?) e, a volte, anche noi siamo stati colti dal dubbio mentre giocavamo allo impiccato.
D'accordo, siete dei mostri di sapienza in geografia. Non ci resta che incoraggiarvi a tentare di sostituire i nomi di città italiane con nomi, per esempio di città straniere, nomi di personaggi, o qualsiasi cosa vi venga in mente. Vedrete che questo programma potrà diventare abbastanza interessante e, perché no, educativo.
Per cambiare il programma dovete soltanto fissare la variabile NP = 82 al nuovo valore necessario a contenere tutti i nuovi nomi che scriverete nei DATA riguardanti i nomi di città e, naturalmente, cambiare la serie dei DATA.
Tutto questo, certo, nei giorni piovosi.

PAOLO CIVARDI

GIOCHIAMO CON IL TI 99/4A

Un libro che non deve assolutamente mancare nella vostra libreria.

Potrete riceverlo a casa vostra a sole **L. 9500** + Spese di spedizione.

Per ogni richiesta di contrassegno, utilizzate il tagliando qui sotto spedendolo a:

NEWSOFT snc
via S. Jacini, 4
20121 MILANO

Vogliate spedirmi in contrassegno di L. 9500 + spese il libro: «GIOCHIAMO CON IL TI 99/4A»

Cognome .....

Nome .....

Indirizzo .....

.....

Cap. ..... Loc. .....

COME OTTENERE IL MULTICOLOR MODE

Il multicolor ha la particolare caratteristica di gestire uno schermo, composto da 48 righe per 64 colonne, esattamente il doppio di quelle gestite nel normale modo grafico.
Infatti il computer divide lo schermo in 3072 quadratini, ognuno dei quali, può avere un colore diverso, senza alcuna limitazione.
Un fatto alquanto negativo è che questi quadratini non sono definibili in alcun modo. Questo dipende dal fatto che il computer considera ognuno di questi quadratini come un solo elemento. In pratica li considera come pixel singolarmente indirizzabili.
Per poter ottenere il multicolor mede, dobbiamo intervenire sui registri della VDP, cambiandone il contenuto.
Il registro della VDP sul quale dobbiamo intervenire è il numero 1. In esso, dobbiamo inserire il valore esadecimale >E8, che tradotto in decimale risulta essere 232.
Se si opera in assembler, cioè scrivendo direttamente il programma con un compilatore, dobbiamo fare attenzione, perchè detto valore oltre che essere scritto nel registro 1 della VDP va scritto anche nella copia dei registri VDP alla locazione esadecimale >83D4 corrispondente in decimale a — 31788.
Si dovrà operare nel seguente modo:

```
LI RO, >01E8
BLWP C δ>VWTR
SWPB RO
MOVB RO, δ>83D4
```

In extendede basic con l'espansione di memoria, le cose si semplificano molto. Per entrare in multicolor è sufficiente eseguire la CALL LOAD:

```
CALL LOAD (-31788,232)
```

Per uscire dal suddetto modo grafico battere alla cieca la CALL LOAD:

```
CALL LOAD (— 31788,224)
```

Ciò è dovuto dal fatto che l'extended basic prende i valori direttamente dalla copia dei registri, quindi risulta più facile l'inserimento in uno di essi.
Tuttavia questa semplicità comporta anche dei lati negativi, perchè in extended risulta pressoché impossibile gestire in modo soddisfacente il multicolor.
Le ragioni sono varie, la principale è che il modulo extended basic resetta tutti i puntatori dello schermo, dei caratteri e dei colori ogni volta che gli viene passato il controllo.
Dato che il multicolor ha bisogno per poter funzionare correttamente di valori di schermo e colori totalmente diversi da quelli del modulo, risulta impossibile, o per meglio dire difficoltoso, fare un programma assembler che interagisca con il basic del modulo.
Nelle altre puntate, ci siamo sempre sforzati di darvi una prova pratica di quanto si discuteva nell'articolo, ma data la complessità di tale modo grafico e data la scarsa utilità pratica, abbiamo preferito optare per altre routine di maggiore interesse.
Potete comunque provare ad eseguire le CALL LOAD sopra citate per farvi un'idea del tipo di capacità del nostro computer.
La routine che vi proponiamo questo mese nella rubrica software extended +32 K ram, può a nostro parere riscuotere molto successo.
La routine in lm, vi permetterà di operare degli scroll video verso il basso e verso sinistra.
Per maggiori informazioni a questo riguardo guardate le pagine della rivista dedicate alla rubrica software.

EZIO MONTINI

SCROL DEL VIDEO

Eccovi un'altra routine in linguaggio macchina che vi servirà moltissimo nei vostri giochi.
Si tratta di due sottoprogrammi che permettono, se mandati in esecuzione, di eseguire uno scrol del video verso sinistra SLEFT, o verso il basso SDOWN.
Il programma assembli come potete vedere dal lisatto dell'oggetto-sorgente, è composto da due punti di entrata SDOWN, SLEFT e un sottoprogramma di utilizzo interno.
Detto sottoprogramma chiamato CANC ha il compito di cancellare le impurità che lo scrol provoca all'interno dello schermo, mantenendo il video sempre pulito.
Per mandare in esecuzione i sottoprogrammi da basic ''EXTENDED + 32 K'', dobbiamo prima caricare e mandare in esecuzione il programma basic che trovate nella cassetta chiamato SCROL VIDEO.
Solo dopo aver eseguito questa operazione possiamo fare uso delle nuove istruzioni.
Possiamo usare due nuove istruzioni, rispettivamente SLEFT e SDOWN.
Per mandarle in esecuzione è sufficiente scrivere:
1) CALL LINK (''SDOWN'')
2) CALL LINK (''SLEFT'')
Ognuna di esse è completamente indipendente dall'altra e può essere eseguita sia in modo programma che in modo comando.
Per evitare problemi di caricamento da cassetta, e data le relative brevità del programma basic che crea i comandi, pubblichiamo oltre alla sorgente e l'oggetto del programma, anche il programma basic.
Ora sta a voi fare fruttare al meglio queste nuove routines, che aprono al mondo della programmazione dei giochi, assieme a EXPANDER nuove ed interessanti possibilità.
Nei prossimi numeri vi spiegheremo in che modo sia possibile collegare due programmi assembler separati es. EXPANDER con SCROL ecc.
Vi informiamo che sono in fase di progettazione anche le routine che permettono lo scrol del video nelle rimanenti direzioni, e verranno pubblicate nei prossimi numeri. Vi auguriamo buon lavoro e se riuscite a fare qualche programma di una certa qualità che utilizzi le nostre routines, speditelo in redazione.

EZIO MONTINI

AVETE ACQUISTATO QUESTO NUMERO?

''TI 99 NEWSOFT''
è in edicola
dal giugno 1984
Se desiderate
ricevere a casa
gli arretrati,
scriveteci!

Ve li invieremo
in contrassegno
a Lit. 15.000
+ spese cadauno.

TI 99 NEWSOFT
Via S. Jacini, 4
20121 MILANO

ADVENTURE NOW

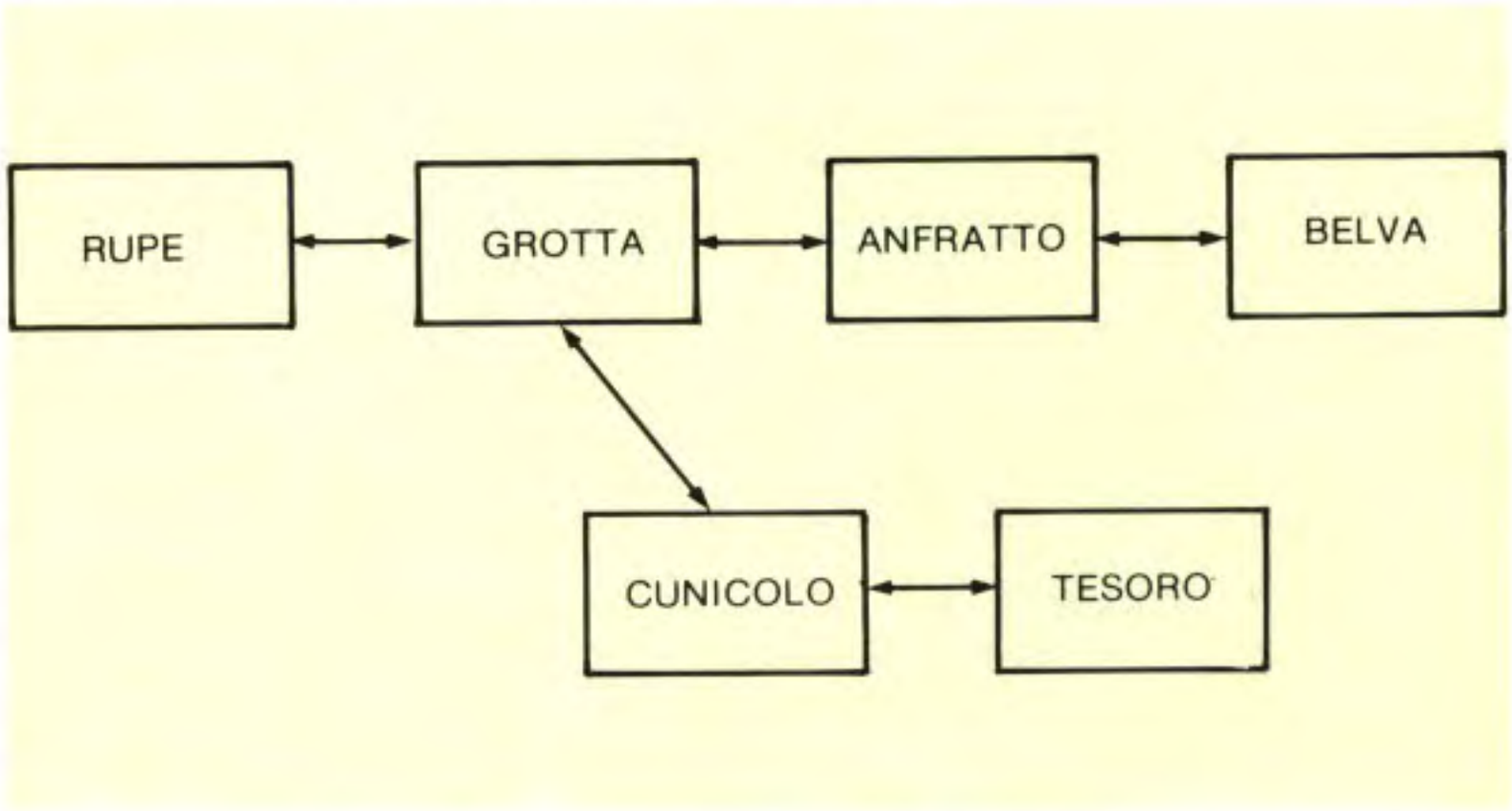
Finalmente parliamo di ADVENTURE.
Per coloro che non lo sapessero ancora e per quelli che l'avessero dimenticato ricordiamo che non si tratta di giochi per i quali necessitano riflessi eccezionali e nervi d'acciaio (come per gli arcade game), ma di giochi che vi costringono ad esplorare luoghi misteriosi, grotte, castelli, lande desolate. Gli unici requisiti sono: concentrazione, logica, creatività e naturalmente fantasia. Appartengono a questa schiera (e non è pubblicità!!!) l'HOBBIT, THE DALLAS QUEST, DUNGEON & DRAGONS, RETURN TO PIRATE'S ISLE (per TI 99). A proposito di quest'ultimo consigliamo a tutti di acquistarlo (Fantastica la grafica!!) ed ai pochi che l'hanno imploriamo di scriverci poiché noi siamo giunti ad un punto morto.
Chiudiamo qui la parentesi e passiamo a un po' di storia. Il «padre» di questo tipo di giochi, cioè colui che per primo scrisse uno di questi giochi è da tutti considerato Scott Adams che scrisse un'avventura per il TRS 80 con 16 Kb di memoria. Da allora molta strada è stata fatta, gli adventure hanno conquistato tutto il mondo, specialmente i paesi anglosassoni, e il signor Scott Adams ha una software-house tutta sua. Vediamo ora di chiarire la posizione del computer in questi giochi.
Dicevamo prima che gli adventure consistono nell'esplorazione di luoghi fantastici e nella ricerca di favolosi tesori. Bene, il compito del computer è quello di fornire tutte le informazioni su ciò che ci circonda e di aiutarci in caso di difficoltà. Per farla breve il computer si comporta come un Robot: vediamo con i suoi occhi, camminiamo con le sue gambe, sentiamo con le sue orecchie, prendiamo con le sue mani e così via. Quindi, riassumendo, il computer ci fornirà la descrizione del luogo in cui ci troviamo, l'elenco degli oggetti che ci stano intorno (se ce ne sono), qualche informazione sul percorso da intraprendere e, se possibile, un'immagine grafica in modo da renderci meglio conto della situazione. Adesso che abbiamo spiegato cosa è un adventure, passiamo ad esaminare due punti salienti:
1 — Come si gioca
2 — Come si costruisce un modello standard per scrivere gli adventure.

1 — Spieghiamo subito come si usa un adventure così da non perdere più tempo in futuro.

Dunque, per giocare si impartiscono al computer dei semplici comandi (di solito di 1 o 2 parole) usando logica e fantasia. Ad esempio per dire VAI A NORD è sufficiente digitare sulla tastiera NORD, per prendere un oggetto basta scrivere PRENDI...

Come avrete senz'altro capito non c'è molto da dire, ma più che altro da pensare...
Purtroppo la maggior parte degli adventure in commercio sono in lingua inglese che non tutti conoscono e ciò costituisce un freno alla loro diffusione in Italia.
2 — Ed ora passiamo ed esaminare il punto che a noi dell'MMG interessa maggior-

mente mostrarvi come si costruisce uno schema fisso in modo da poter scrivere avventure diverse facendo solo piccole modifiche al programma principale. Inoltre se ogni volta ci presterete la dovuta attenzione vedrete che alla fine avrete raggiunto una notevole conoscenza del BASIC. Infatti questo tipo di Games sfrutta al massimo tutte le potenzialità del BASIC.
Analizzeremo, d'ora in poi, una componente per volta in modo da avere ben chiara la funzione svolta da ogni singola routine del programma. Inoltre per comodità di linguaggio, chiameremo ogni luogo «stanza» e, sempre per nostra comodità, assegneremo ad ogni stanza un numero (vedremo in seguito a cosa servirà). La prima cosa da fare è di stabilire quali saranno i luoghi che interesseranno la nostra avventura, un filo logico che li colleghi insieme. Fatto ciò potremo iniziare a stendere sulla carta una specie di mappa topografica dell'avventura (da migliorare eventualmente in seguito).
Cercherò di spiegarmi meglio con un semplice esempio:
Supponiamo che la nostra avventura si svolga in un territorio montano; i luoghi che ci interessano saranno RUPI, GROTTE, ANFRATTI, CUNICOLI...
Per quanto riguarda la parte logica immagineremo di trovarci su una rupe, di trovare una grotta in cui vi è una biforcazione: 2 strade. La prima conduce alla tana di una belva e quindi a morte certa, l'altra attraverso un cunicolo ad un anfratto in cui vi è un tesoro. Come detto sopra ora abbozziamo una specie di mappa:



Le frecce indicano le possibili direzioni da intraprendere. Notate sempre il numero associato ad ogni stanza. Chiaramente questa è solo una rappresentazione semplicistica: potremo però renderla più impegnativa con piccole modifiche. Infatti potremo pensare di avere la possibilità di combattere contro la belva e, in caso di vittoria, aprire un passaggio segreto per giungere al tesoro attraverso la stanza 4. Oppure di trovare il buio più assoluto nella grotta e di doverla illuminare con una torcia trovata prima sulla rupe. Come vedete le possibilità sono molteplici e dipendono unicamente dalla vostra fantasia. A questo punto il nostro micro-adventure ha già preso un'altra «piega» e per risolverlo è necessario superare almeno uno dei due ostacoli (illuminare la grotta per vedere i due passaggi).
Un adventure vero e proprio è costituito da un'infinità di ostacoli, trabocchetti, situazioni particolari e rebus da risolvere per giungere alla meta prefissata; da notare inoltre che spesso non esiste un'unica via per arrivare alla meta, ma, come nel nostro esempio, diverse con ostacoli di differenti difficoltà. Per questa volta ci fermiamo qua. Speriamo di avervi fornito una panoramica sufficientemente vasta del mondo degli adventures e soprattutto di non avervi annoiato troppo. Quindi dopo la lettura dell'articolo vi invitiamo a cimentarvi con l'avventura della cassetta.

DANIELE CATALFAMO

DISCOVERY

Si tratta di una semplice avventura, scritta in TI BASIC, che vi trascinerà nel lontano mondo del Far-West.
Lo scopo è di riuscire ad attraversare una strada sorvegliata da due feroci pistoleri restando indenni. Per eludere la loro sorveglianza dovrete trovare due oggetti che vi permetteranno di passare inosservati evitando i loro mortali proiettili.
L'avventura è, chiaramente, tutta in italiano e, per questa volta, vi forniremo l'intero vocabolario dei verbi accettati dal computer (vedi fig. 1). Per impartire i comandi basterà digitare ogni volta VERBO+OGGETTO e battere ENTER. Inoltre per ogni parola sarà sufficiente digitare solo 4 lettere, quindi per dire PRENDI LA COPPA basterà digitare **PREN COPP** e premere ENTER.
Detto questo, detto tutto. Penso che dovreste essere tutti in grado di giungere sino in fondo all'avventura. Comunque per i pochi che si ritroveranno a brancolare nel buio, appuntamento il prossimo mese con la soluzione dell'avventura e la seconda parte di ADVENTURE NOW.
N.B. L'avventura presentata è una conversione per TI 99 di un programma per CBM, ed è stata scritta piuttosto velocemente, per cui se listerete il programma (per darci una sbirciatina!!) consigliamo di non adottare lo schema usato nell'adventure in questione.

DANIELE CATALFAMO

VERBI ACCETTATI

Prendi: quando vuoi prendere qualcosa che vedi
Posa/Getta: quando vuoi lasciare qualcosa che possiedi
Tocca: per tastare qualcosa
Vai/Corri: per andare in qualche direzione
Taglia
Sega
Abbatti
Metti/Infila
Unisci/Connetti

Inventario

Inoltre per impartire i seguenti comandi basta digitare una sola lettera e battere ENTER.

N: per andare a NORD
S: per andare a SUD
E: per andare a EST
O: per andare a OVEST
G: per guardare intorno
I: per l'inventario
F: per finire (se proprio non ce la fate più)

TREDIMENSIONI

Solidi in prospettiva con il TI 99/4A

Come dice il nome questo programma serve a visualizzare in prospettiva dei solidi. Non ci dilunghiamo con delle spiegazioni su come vengono effettuati i calcoli della prospettiva, basti sapere che si fa riferimento a un sistema di assi coordinati con il centro al centro del teleschermo per cui potrete dare dei valori sia positivi che negativi per le coordinate. Vedrete il solido come se vi trovaste sull'asse delle X e ad un'altezza che potrete segliere di volta in volta.

Passiamo ora ad esaminare le possibilità operative del programma. Appena dato il RUN apparirà il menù principale con la scelta delle opzioni esaminiamole una per una:

1. *Definizione punti.*

Con questa opzione si possono caricare in memoria le coordinate dei vertici della nostra figura. Prima di iniziare a inserire i dati occorre indicare da quale vertice iniziate, dopodiché potrete inserire in ordine le coordinate X, Y, Z dei singoli vertici.

2. *Definizione linee.*

Dopo aver inserito le coordinate dei vertici occorre indicare al computer quali di essi devono essere congiunti. Occorre anche qui indicare lo spigolo di partenza quindi inserire di volta in volta il vertice di partenza e quello di arrivo dello spigolo (ad es. 1 - 2 indica lo spigolo che parte dal punto 1 e arriva al punto 2 definiti nell'opzione 1).

3. *Rotazione.*

Come dice il nome stesso serve a far ruotare delle figure. Per prima cosa occorrerà indicare gli angoli di rotazione (in gradi) rispetto a ciascun asse cartesiano quindi come in molte altre sezioni occorre indicare i punti sui quali si vuole effettuare le modifiche cosa che alla lunga può essere noiosa però è introdotta poiché può capitare di avere più figure in memoria e di voler agire su una sola o su una parte di essa.

4. *Cambiamento scala.*

Questa opzione serva per ingrandire o per rimpicciolire una figura, occorre inserire il fattore di ingrandimento (se è maggiore di uno la figura viene ingrandita, se minore rimpicciolita) quindi i punti sui quali si vuole agire.

5. *Distorcere una dimensione.*

Come la precedente però agisce selettivamente sulla dimensione desiderata.

6. *Trasferire punti.*

Serve a fare delle copie dei punti desiderati; ad esempio possiamo copiare il punto 1 nel punto 5. Occorre inserire il punto di partenza da copiare e quello in cui deve essere copiato e quindi il numero di punti da trasferire. Questa opzione può servire per fare delle copie di una figura da trasferire poi in un altro punto dello schermo con l'opzione 8.

7. *Spostamento punti.*

Serve per modificare le coordinate di un singolo punto. Inserire prima il punto da modificare poi di quanto si vogliono incrementare le singole coordinate.

8. *Spostamento figura.*

Serve a spostare delle figure sullo schermo, bisogna prima indicare lo spostamento per ogni singola coordinata poi come al solito quali punti

si vuole spostare.

9. *Lista punti.*
Come dice il nome stesso mediante questa opzione si possono vedere le coordinate dei punti in memoria.

10. *Lista linee.*
Come la precedente però serve per listare gli spigoli che sono stati definiti (per continuare a vedere i punti successivi basta premere S quando compare la scritta "SCROLL?" Se si preme un altro tasto si torna al menù).

11 - 12. Come facilmente si può capire servono rispettivamente a salvare e a caricare i dati di una figura dal registratore a cassette.

13. *Visualizzazione.*
Questa è sicuramente la sezione più importante e permette la visione in prospettiva della figura memorizzata. È necessario introdurre la distanza del punto di fuga che non deve trovarsi all'interno della figura (più il punto di fuga è vicino più la figura appare grande e la prospettiva accentuata) quindi occorre inserire l'altezza dell'osservatore rispetto al piano xy che ha altezza zero.
Di questo programma ne ho fatte due versioni, una per chi possiede il solo modulo EXTENDED BASIC e l'altra per chi possiede anche l'espansione di memoria.
La versione per l'espansione di memoria ha bisogno di un programma in linguaggio macchina per la gestione dell'alta risoluzione, tale programma è formato da diverse subroutine utilizzabili anche in altri programmi e la cui spiegazione è fornita a parte. Nella cassetta viene fornito anche un esempio da caricare con l'opzione 12, si tratta di un parallelepipedo che potrete vedere richiamando poi l'opzione 13, dando come distanza del punto di fuga 200 e 30 come altezza dell'osservatore.
Passiamo ora ad esaminare il programma per il solo EXTENDED BASIC; comunque l'altro è perfettamente uguale tranne nella sezione di visualizzazione della figura.
Poiché l'EXTENDED BASIC non dispone di una grafica in alta risoluzione si è dovuti ricorrere a una funzione di PLOT in BASIC per cui la velocità di tracciatura delle figure non è certo eccezionale, inoltre non è possibile tracciare figure molto grandi in quanto vi è la limitazione dei 111 caratteri definibili dall'utente. Tutte queste limitazioni sono state superate nella versione per l'espansione utilizzando il modo BIT-MAP gestito da routine in linguaggio macchina.

M.M.G.
GIAMPAOLO BOTTIN

100 - 150	inizializzazione e stampa menù
160 - 170	scelta opzione
180 - 250	definizione punti
260 - 290	spostamento punti
300 - 310	routine per il calcolo del centro di una figura
320 - 370	spostamento figura
380 - 420	cambiamento scala
430 - 510	distorsione di una dimensione
520 - 580	rotazioni
590 - 630	trasferimento punti
640 - 750	visualizzazione
760 - 800	lista punti
810 - 860	lista linee
870 - 930	definizione linee
940 - 1020	salvataggio dati
1030 - 1070	caricamento dati
1080 - 1160	routine di LINE per tracciare gli spigoli
1170 - 1230	routine di PLOT

Principali variabili utilizzate:

C (50)	coordinate X delle figure memorizzate
D (50)	coordinate Y
E (50)	coordinate Z
H (50), G (50)	contengono le coordinate per la stampa sullo schermo delle figure.
L(110,2)	contiene gli spigoli della figura
DM	numero massimo di spigoli memorizzabili
C3,D3,E3	coordinate centro figura
X3,Y3	coordinate del centro dello schermo.

BIT-MAP MODE

Queste routine in linguaggio macchina servono alla gestione del modo grafico in BIT MAP di cui si è già parlato su queste pagine. Questi sottoprogrammi occupano poco più di 1 Kb nella low memory e vengono caricate in memoria da cassette utilizzando un LOADER caricato precedentemente da un programma basic. I sottoprogrammi per utilizzare il registratore a cassette sono:

CALL LINK («BSAVE», IND1, IND2)
salva su cassetta il contenuto della CPU RAM compresa tra IND1 e IND2 che devono essere specificati in DECIMALE.

CALL LINK («BLOAD»)
carica una zona di memoria precedentemente salvata (con questo comando vengono caricate le routine per l'altra risoluzione)
N.B.
Il LOADER viene caricato nella high-memory per cui bisogna fare attenzione che il programma basic lasci liberi alcune centinaia di bytes per evitare pericolose sovrapposizioni.

Gestione BIT-MAP MODE
Prima di utilizzare l'alta risoluzione in un programma occorre inserire la seguente istruzione per limitare la VDP ram disponibile per l'interprete BASIC:

CALL LOAD (-32658,56,0"",-32732,55,248)
rimangono così liberi circa 2Kb per le strighe e per i nomi delle variabili

CALL LINK («HGR», col)
inizializza i registri VDP e la pagina grafica colorando lo sfondo del colore specificato nella CALL LINK

CALL LINK («PLOT»,Y,X-,COL)
accende il pixel alle coordinate Y (riga) e X (colonna) e lo colora del colore COL

CALL LINK («LINE», Y1, X1, Y2, X2, COL)
traccia un segmento dal punto di coordinate Y1,X1 al punto y2,X2 e di colore COL

CALL LINK («DRAW», Iy,IX-,COL,...)
traccia un segmento a partire dall'ultimo punto plottato (ad esempio con una CALL LINK («PLOT»,Y,X,C)) al punto ottenuto sommando IY e IX alla coordinate di tale punto. Naturalmente Col indica il colore del segmento. La terna di valori può essere ripetuta più volte permettendo così di tracciare delle spezzate con una sola istruzione, ad esempio:
CALL LINK («PLOT», 100, 100, 4)
CALL LINK («DRAW», -50, 0, 4, 0, 50, 4, 50, 0, 4, 0, -50, 4)
traccia un quadrato verde
CALL LINK («CIRCLE», Y, X, RY, RX, COL)
traccia una ellissi di centro Y, X, con semiasse verticale uguale a RY e semiasse orizzontale uguale a RX e di colore COL naturalmente se i due semiassi sono uguali verrà tracciata una circonferenza
CALL LINK («FILL», Y, X, COL)
permette di riempire aree chiuse con il colore COL, è sufficiente che il punto Y, X si trovi all'interno dell'area da riempire. Attenzione è sufficiente un buco di un solo pixel e invece di limitarsi a riempire la zona desiderata verà riempito tutto lo schermo
CALL LINK («N»)
permette di riportare alla nor-

malità lo schermo alla fine del programma
CALL LINK («POINT», Y, X, A)
permette di testare lo stato di un pixel A=0 se il pixel è spento A=-1 se il pixel alle coordinate Y, X è acceso
CALL LINK («POKEV», IND, A, B,...)
permette di scrivere nelle VDP ram all'indirizzo IND i valori indicati nella CALL LINK
CALL LINK («PEEKV», IND, A, B,...)
permette di leggere bytes nella VDP ram all'indirizzo IND e li assegna alle variabili della CALL LINK.
Se l'indirizzo è negativo si modificano i registri VDP.
N.B.
L'origine degli assi è posta in basso a sinistra per cui lo schermo appare come il primo quadrante di un sistema di assi cartesiani. Le coordinate verticali (Y) vanno da 0 a 191 quelle orizzontali (X) da 0 a 255 mentre i colori sono quelli soliti del BASIC.

GIAMPAOLO BOTTIN

Questo mese il "**TI 99 NEWSOFT**" compie un anno.
Quale modo migliore di augurargli lunga vita
se non quello di abbonarsi?
Con sole L. 95.000 riceverete
11 numeri della rivista e in più avrete diritto
agli sconti sul software e l'hardware offerto da

COMPUMARKET

Per abbonarsi basta inviare
un assegno non trasferibile o un vaglia postale a
NEWSOFT snc - Via S. Jacini, 4 - 20121 Milano

GOLF

«GOLF» è un gioco scritto per il TI 99/4A in versione base ed è concepito per un solo giocatore.
Lo scopo del gioco è, come nella realtà, quello di riuscire a completare il percorso mandando in buca la pallina e totalizzando il massimo punteggio.

Il GIOCO

Caricato il programma e dato il RUN, apparirà sullo schermo il titolo del gioco unitamente alle opzioni per G (giocare) oppure per avere le I (istruzioni). Se scegliete quest'ultima, appariranno sul video le poche regole da seguire per tentare di giungere all'ultima buca cosa che, come vedrete, si rivelerà certo possibile ma altrettanto certamente non facile.
Lette le istruzioni premete un tasto qualsiasi: in questo modo inizierete il vostro percorso sotto l'occhio attento degli spettatori sulle tribune, pronti ad osannarvi se il vostro tiro finirà in buca.
Al centro delle tribune un tabellone luminoso vi informerà sul record da battere, sul numero di tiro che state effettuando (avete a disposizione 4 tiri per ogni frazione di percorso), sul numero di buca alla quale state cimentandovi e, naturalmente, sul vostro punteggio.
Ciò che voi dovrete decidere ora sarà, innanzitutto, il tipo di parabola che la pallina dovrà compiere per superare gli ostacoli (muri, cespugli, alberi, laghetti, ecc.): usate il tasto — 1 — per un tiro alto e corto, il tasto — 2 — per un tiro normale e il tasto — 3 — per un tiro basso e lungo. Ovviamente dovrete provare i vari tipi di tiro per comprendere il significato di «alto», «basso», «lungo», ecc..
Dopo questa prima decisione premete il tasto prescelto per un breve periodo (cominciate con un secondo circa): nel momento stesso in cui rilascerete il tasto la vostra pallina si innalzerà, lasciando sullo schermo la traccia della sua parabola.
Se durante il volo o nel toccar terra urterà gli ostacoli disseminati lungo il percorso, la pallina la perderete: di questo verrete avvisati con un messaggio, come pure dei tiri che ancora avrete a disposizione. Ricordate che NON dovete superare la buca, pena la perdita della pallina: potete invece raggiungerla in più colpi successivi. Se non ci riuscirete nei 4 tiri a vostra disposizione sarete, ahimè, squalificati.
In quest'ultima evenienza lo schermo verrà pulito e apparirà l'indicazione del record,

VARIABILE USATE:

RCD	=	contiene il valore del Record
TR	=	contiene il numero del tiro
BC	=	contiene il numero della buca
PT	=	contiene il punteggio totalizzato
X	=	colonna da cui inizia la parabola
R	=	riga rappresentante il suolo
N	=	colonne di distanza da X del tiro effettuato
KK	=	fattore di correzione altezza parabola
Y	=	valore di altezza pallina
C	=	colonna contenente la buca
V	=	tiri residui

del vostro punteggio e il menù per scegliere se continuare a giocare o finire o avere le istruzioni.
Se scegliete di giocare ancora, ricomincerete dalla buca n° 1.
Ogni volta che la vostra pallina finirà in buca avrete diritto a giocare alla successiva buca: verrà pertanto visualizzato un nuovo percorso affinché possiate cimentarvi ulteriormente.
Se riuscirete a completare anche la 9° buca (tante sono infatti le frazioni di percorso) ebbene... riceverete il dovuto riconoscimento a tanta abilità!
Ricordate che il vostro punteggio sarà tanto più alto quanti meno tiri effettuerete per giungere in buca e, inoltre, esso aumenta con l'aumentare della difficoltà dei percorsi.
Un'ultima osservazione: in lugo del tasto — 2 — potete usare un qualsiasi altro tasto ad eccezione, ovviamente di — 1 — e — 3 —.

GIAMPIERO CALOVI

100 - 170	Definisce lo schermo iniziale di attesa;
180 - 280	Definisce i colori dei caratteri utilizzati;
290 - 480	Definisce i caratteri usati nel gioco. Il carattere 93 (parentesi quadra chiusa) dovrà essere usato, dove previsto, per garantire scritte su fondo nero, senza discontinuità di colore.
490 - 650	Definisce la parte superiore dello schermo contenente gli spettatori, la tribuna e il tabellone luminoso. In queste PRINT gli SPAZI sono sostituiti dal caratt. 93
660 - 670	Rimanda alla linea corrispondente al percorso da generare nella parte inferiore dello schermo.
680 - 1020	Questo blocco di istruzione calcola e traccia la traiettoria della pallina: vale la pena di spendere alcune righe per capire come funziona. Quando il computer è in attesa del vostro tiro è fermo alla linea 840 - 850, in quanto la variabile di stato S è 0. Nel momento in cui voi premete un tasto la variabile S diviene diversa da 0 e quindi il programma prosegue nell'esecuzione della linea successiva. Questa linea assegna alla variabile un valore diverso a seconda che voi abbiate premuto il tasto — 1 —, — 3 — o diverso da questi. Ho preferito adottare questa soluzione in quanto consente di evitare gli IF K= ... THEN... e i GOTO inevitabilmente necessari per «saltare» certe linee. Immaginate di aver premuto il tasto — 1 — il cui codice ASCII è 49: risulterà quindi vera l'uguaglianza K=49 e, di conseguenza, false le altre uguaglianze. Siccome il «vero» valore — 1 — e il «falso» vale 0 il prodotto del primo termine risulterà 2.5 (cioè — 2.5 x (-1)) mentre gli altri termini saranno uguali a 0, in quanto risultato del prodotto di un numero per un «falso» (cioè 0). La successiva linea 880 è ancora una CALL KEY: se il tasto risulterà ancora premuto la linea 890 verrà superata, N sarà incrementato di una unità e quindi la linea 910 rimanderà il programma alla linea 880. Se stavolta il tasto risulterà rilasciato S sarà = 0 e quindi il programma salterà alla linea 920 dove verrà calcolato, colonna per colonna, il valore Y, che determina di quanto si è sollevata la pallina. Quindi il programma legge il carattere esistente nel punto dello schermo dove dovrà passare la pallina e se risulterà «valido» stamperà il carattere per disegnare la traiettoria altrimenti porrà fine al tiro. Verrà quindi verificato se il valore di N è uguale, maggiore o minore di C (colonna della buca) e il programma sarà rimandato alle linee 2780, 2710, 2430 per il proseguimento o il termine del gioco.
1030 - 2390	Stampa la parte inferiore dei nove schermi rappresentati i percorsi. Purtroppo, non disponendo di una stampante, non so in che modo verranno riprodotti i caratteri speciali.
2400 - 2690	Questa parte stampa i messaggi inerenti ai tiri non validi e decide la prosecuzione o il termine del gioco.
2700 - 2760	Questa parte aggiorna le variabili TR e X e vi permette di proseguire nei vostri tiri.
2770 - 2970	Questa parte genera gli effetti ottico-sonori, aggiorna le variabili PT, RCD, TR, BC e vi permette di passare allo schermo successivo.
2980 - 3020	Routine per la stampa messaggi.
3030 - 3050	Suono per pallina in buca.
3060 - 3100	Suono per tiro non valido.
3110 - 3340	Schermo di fine gioco.
3350 - 3650	Schermo «di premio» per coloro che riusciranno a completare il nono schermo.
3360 - 3890	Contiene il motivo musicale che accompagna lo schermo «premio».
3900 - 4130	Contiene lo schermo di presentazione: ancha qui il carattere speciale usato (129) non sarà probabilmente riprodotto da una stampante.
4140 - 4360	Contiene le istruzioni del gioco: fate attenzione che gli spazi tra le parole sono sostituiti dal carattere 93 (parentesi quadra chiusa).

EASY COCKTAILS

Il programma che vi presentiamo questa volta non è il solito gioco. Si tratta di qualcosa di diverso. Infatti il nostro TI 99 si trasformerà in un abilissimo BARMAN, che avrà il compito di aiutarvi a preparare ''gustosissimi cocktails''. Con il nome di cocktail si intende generalmente una bevanda ottenuta dalla miscelazione di liquori diversi. Il programma prevede la possibilità di preparare 4 tipi diversi di queste bevande:
Aperitivi, digestivi, long drinks, notturni.

APERITIVI

I cocktails aperitivo possiedono la caratteristica di essere preparati con facilità. Sfruttano inoltre liquori ed altri ingredienti che stuzzicano l'appetito, e danno un tocco di allegria e raffinatezza che fa sempre piacere. Il successo di questa bevanda, - quindi -, consiste proprio nell'immediatezza della loro preparazione e nella possibilità di servirli con moltissimi ''stuzzichini'' che rendono sempre un po' *chic* tutta l'atmosfera.

I DIGESTIVI

Questi, al contrario degli aperitivi, occupano la parte finale di un pasto e sono un elemento necessario ed indispensabile dopo un abbondante ''abbuffata''. Sono composti in genere principalmente da un SUPER alcoolico con l'aggiunta di succhi e altri liquori aromatizzanti. Possiedono la caratteristica di aiutare e facilitare la digestione, per cui oltre ad essere gradevoli sono piuttosto utili. Non richiedono una eccessiva elaborazione durante la fase di preparazione e possono essere gustati durante il pomeriggio o nel corso della prima serata.

I LONG DRINKS

Queste bevande occupano una posizione di rilievo tra tutti i cocktails. Sono caratterizzati da una grande varietà di colori e da una ricercata decorazione, in genere a base di frutta fresca ed esotica. Sono inoltre molto dissetanti ed adatti così alle stagioni più calde. Per la preparazione si utilizzano, di solito, succhi di frutta in modo che, oltre ad essere particolarmente dissetanti sono anche poco alcoolici. La loro decorazione permette a chiunque di sbizzarrirsi come crede e di sfruttare al massimo la propria fantasia. Pezzi di frutta, cannucce coloratissime, ombrellini. ecc. sono tra gli accessori più utilizzati.

I NOTTURNI

Ma il momento migliore per gustare i cocktails è forse la sera. Ecco quindi l'esigenza di creare un tipo particolare di

cocktails, indicati appunto per questo momento della giornata. Sono i cosiddetti ''NOTTURNI'', si distinguono per la particolare attenzione prestata durante la loro preparazione e presentazione.

ACCESSORI INDISPENSABILI

Il primo tra tutti gli ''attrezzi'' necessari è probabilmente il MIXING GLASS, o bicchiere miscelatore. Possiede una forma conica, (il più ricercato è di cristallo e fatto in modo che il ghiaccio non trabocchi). In dotazione viene fornito un LUNGO CUCCHIAIO che ha il compito di mescolare le bevande.
Non possiamo poi dimenticare lo SHAKER, composto da 2 pezzi che si incastrano l'uno nell'altro e che serve appunto ad agitare energicamente i vari ingredienti.
Di notevole utilità si rivela anche il MISURINO o anche BICCHIERINO GRADUATO, che vi permette di dosare gli ingredienti in maniera perfetta.
Dovrete poi possedere una vasta collezione di quegli accessori che rendono un cocktail invitante, sto parlando di cannucce, spilloni, ombrellini, stecchini, sottobicchieri, vassoi. Per contenere gli stuzzichini, ossia olive, salatini e via dicendo.
Per quanto riguarda il ghiaccio, è doveroso fare un discorso a parte. Infatti, la cosa migliore, sarebbe usare il cosiddetto *Ghiaccio Cristallino* (G.C.), che ha la proprietà di scioglersi molto lentamente, raffreddando così, senza annacquare la nostra bevanda. Purtroppo questo tipo di ghiaccio non può essere fabbricato da un normale freezer, ma solo da una apposita macchina ''fabbricaghiaccio''.

USO DEL PROGRAMMA

Il programma vi offre un ''menù'' ricco di ben 5 opzioni. Le prime 4 sono relative ai rispettivi tipi di cocktails analizzati poco prima: non dovrete far altro che schiacciare il tasto corrispondente alla bevanda di vostro gusto e leggere quindi le modalità per la preparazione.
La quinta opzione invece vi offre la possibilità di comunicare al TI 99 i liquori in vostro possesso e di sapere così quali cocktail poter preparare. Il computer vi fornirà un elenco di liquori a cui voi dovrete rispondere con SI o NO schiacciando rispettivamente i tasti (1) e (0). Fate attenzione poiché il computer è velocissimo a leggere la vostra risposta.
Accanto al nome di ogni cocktail, vi saranno una serie di piccoli bicchieri indicanti la difficoltà di preparazione dello stesso.

1 bicchiere = FACILE
2 bicchieri = RICERCATO
3 bicchieri = PER ESPERTI.

Buon divertimento e attenzione a non ubriacarvi!!!

M.M.G. DANIELE CATALFAMO

ANALISI DEL LISTATO

80 —	105	Fase di inizializzazione
110 —	150	Fase di selezione dell'INPUT
1000 —	1801	Descrizione degli APERITIVI
2000 —	2700	Descrizione dei DIGESTIVI
3000 —	3601	Descrizione dei LONG DRINKS
4000 —	4600	Descrizione dei NOTTURNI
6000 —	8090	Fase di ''help''
10000 —	10004	Linea DATA
31999 —		Riga di selezione
32000 —		Sottoprogramma MENÙ

H$, K$, J$, Y$, A$ = stringhe costanti
2$ (25) Matrice contenente i liquori
N$ (31) Matrice contenente i nomi dei cocktails
T (25) Matrice vero/falso (1/0) usata per l'help
I,J usate nei cicli
C Variabile di selezione
B Indica quanti bicchieri stampare (1,2 o 3).

La **NEWSOFT** promuove il primo censimento nazionale di tutto il software disponibile per il **TI-99/A4**

allo scopo di consentire a tutti i possessori di TI-99 la massima flessibilità di utilizzo del computer secondo le più specifiche esigenze.
Se avete realizzato dei programmi per TI-99 che ritenete interessanti, utili o divertenti e se siete disposti a venderli ad altri ''novantanoviani'', inviateci per ogni programma la scheda inserita al centro della rivista (potete fare delle fotocopie), più una sola copia della scheda informativa. Le schede devono essere compilate in ogni loro parte, possibilmente a macchina o comunque in stampatello; la descrizione del programma deve essere sintetica ma esauriente. Le schede incomplete o illeggibili non potranno essere tenute in considerazione.
Completato il censimento, la **Newsoft** pubblicherà un catalogo di tutto il software valido ricevuto, completo di descrizione di ciascun programma e prezzo di vendita.
A tal fine vi chiediamo un contributo di Lit. 5.000 per la prima scheda e Lit. 2.000 per ogni scheda successiva, da allegare in francobolli, a parziale copertura delle spese di edizione del catalogo, che verrà spedito gratuitamente a tutti coloro che avranno inviato le schede.
Da quel momento i programmi potranno essere richiesti direttamente alla **Newsoft,** che provvederà a girare le ordinazioni ai possessori dei medesimi, rimettendo loro il compenso stabilito, detratto il 30% a copertura delle spese di gestione e spedizione sostenute dalla Newsoft.
Anche se non avete intenzione di proporre dei programmi da voi realizzati, potete comunque partecipare a questa grande iniziativa come acquirenti: compilando ed inviandoci la sola scheda informativa, unitamente a Lit. 5.000 in francobolli, riceverete il catalogo a casa vostra.
Aspettiamo i vostri migliori programmi, il successo di questa iniziativa dipende solo da voi!

OFFERTA SPECIALE

è pronto finalmente il manuale per l'extended basic in italiano!

ORDINATELO CON LA CARTOLINA DEL COMPUMARKET

L. 15.000 (abbonati L. 13.000)

TI-99 CALC

Un versatile tabellone elettronico che vi permette di calcolare rapidamente il vostro bilancio annuale, le spese condominiali, di visualizzare istogrammi e moltissime altre applicazioni.
Il programma gira in Extended Basic e viene fornito completo di un dettagliato manuale di istruzioni e di 4 esempi applicativi su cassetta.
Richiedetelo utilizzando l'apposita cartolina, al prezzo di L. 35.000 (31.000 per i nostri abbonati).

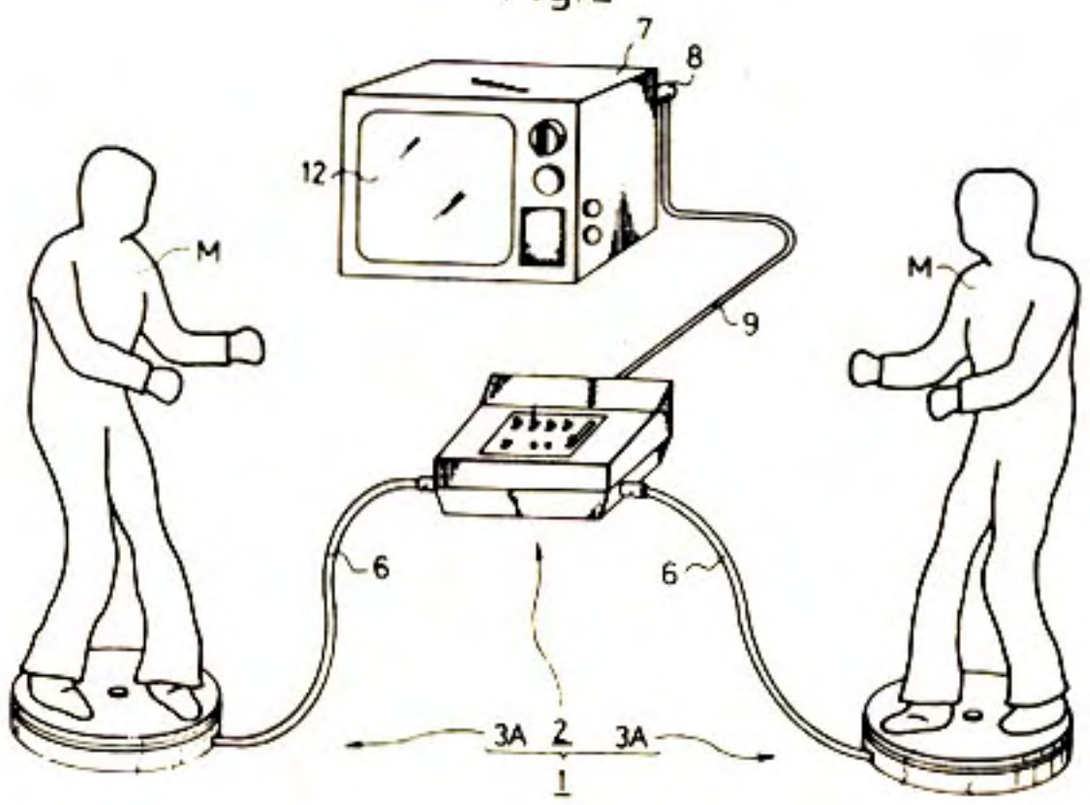
APPARECCHIATURE DI GIOCO ALLA TV

Brevetto NINTENDO 038597

La presente invenzione si riferisce ad un'apparecchiatura di gioco televisivo costituita da un cavo elettrico (6) collegato ad una unità di controllo (3 A) la quale viene manipolata da un giocatore (M), e da un cavo elettrico (9) del segnale di gioco, collegato ad un terminale di antenna (8) di un ricevitore televisivo (7).
Pertanto un qualsiasi gioco viene effettuato dai giocatori i quali cambiano l'immagine sullo schermo TV, manipolando le loro rispettive unità di controllo (3 A).

L'unità di controllo 3 A è un sistema di rivelazione del movimento ed include un elemento mobile che si sposta secondo il movimento di un giocatore ed un elemento di controllo elettrico (non mostrato) che fa variare un parametro elettrico, ad es. una resistenza, a seconda delle variazioni di posizione dell'elemento mobile.
Le variazioni del parametro elettrico provocano variazioni di posizione dei segnali di movimento sullo schermo TV e pertanto consentono l'effettuazione di una molteplicità di giochi, mediante l'uso di un apparecchio televisivo.

Fig. 2



SOFTWARE & Software

L. 3.000

TI-99/4A

IN QUESTO NUMERO:

KOALA

ISTOGRAMMI

ROULETTE

IL GIOCO DEL 15

COMPIUMARKET

SOMMARIO

BASIC pag. 5
Diagrammi a strisce - Un programma di Sergio Borsani che vi darà la possibilità di rappresentare con istogrammi dati di qualsiasi tipo.

EXTENDED BASIC pag. 11
Roulette - Il classico gioco del Casinò. Potete giocarlo con gli amici, ma attenti alla ...camicia.

BASIC pag. 18
Il gioco del 15
Più classico di così si muore.

EXTENDED BASIC pag. 21
Crazy Koala - Una versione per TI-99 di un vecchio videogame da bar. Dovete salvare la vostra Koalina, ma attenti ai massi.

COMPIUMARKET pag. 28

Software & Software
Supplemento al numero 10
di TI 99 NEWSOFT
Registrazione al Tribunale di Milano
numero 323 del 7.7.1984
Direttore responsabile:
ANTONIO C. LOSITO
Coordinatore Editoriale:
VINICIO CIVININI
Direttore Amministrativo:
ENRICO CALLERIO
Art Director:
ROBERTA ROMAGNONI
Illustrazioni:
MARIA TERESA COCOZZA
Segreteria:
CRISTINA FRISON
Redazione:
FIORENZA AURIEMMA
Hanno collaborato al numero:
SERGIO BORSANI
VITTORIO CONSONNI
MARIO LIVRAGHI
MARCO SQUINTANI
PAOLO VENTAFRIDDA
Software & Software
e TI 99 Newsfot sono pubblicazioni della **NEWSOFT snc** - Via Stefano Jacini, 4 - 20121 Milano - Telef. 02/807464 - Telex 324264

COPYRIGHT NEWSOFT snc
TUTTI I DIRITTI RISERVATI
Manoscritti e foto originali anche se non pubblicati, non si restituiscono ed è vietata la riproduzione anche parziale di testi, fotografie e listati.

Pubblicità: AV STUDIO
Via Giuseppe Meda, 23
Telefono 02/8357155
Fotocomposizione:
MULTICOMPOS
Via Michelangelo, 6 - Cassina de' Pecchi (MI) - Tel. 02/9521214
Stampa:
La litotipo srl - Milano
Concessionaria per la distribuzione:
EUROSTAMPA srl - C.so Vittorio Emanuele, 111 - 10123 Torino
Telefono 011/538166-7

L'editore non si assume responsabilità alcuna sul contenuto degli articoli e/o software inviati. Per gli articoli firmati e/o siglati da collaboratori esterni, la redazione si assume la responsabilità prevista dalle leggi sulla stampa.

Tutti possono collaborare a Software & Software, sia con articoli che con programmi. Scriveteci o telefonateci in redazione per avere tutti i ragguagli.
Spedizione in abbonamento postale gruppo III/70%.

RUN...

...in Basic è il comando che fa partire l'esecuzione di un programma.
Per noi della **NEWSOFT** *è la partenza di* **SOFTWARE & SOFTWARE** *una nuova rivista dedicata al TI-99/4A, un computer forse dimenticato ma* **MAI** *abbandonato da chi se ne intende. E come si dice... a buon intenditor poche parole.*

Ma vediamo di presentarci. ''Software & Software'' è una rivista dove troverete listati di programmi, scritti per la maggior parte da altri lettori. A questo punto qualcuno si potrà chiedere: ma quali altri lettori? — Ma i lettori di TI-99 NEWSOFT. no? L'unica rivista con cassetta italiana per il nostro TI-99.

Infatti la **NEWSOFT** *pubblica da tempo ''TI-99 Newsoft'', diffusa specialmente su abbonamento a tutti gli amici che non vogliono separarsi dal loro computer, che è pur sempre un 16 bit ed ha una grafica invidiabile.*
''Software & Software'', quindi, nasce dall'esperienza di una grande casa editrice e, se voi ci darete fiducia, vi dimostrerà che il buon ''vecchio'' TI-99, non è poi da buttare via.

Ti Basic

DIAGRAMMI A STRISCE

Questo programma scritto in TI BASIC, offre il vantaggio di una maggior versatilità rispetto ad altre versioni in quanto consente l'introduzione di dati dipendenti sia da una legge matematica che da una funzione empirica, consente poi la modifica degli stessi in modo selettivo e sequenziale e prevede l'ordinamento dei dati sia secondo l'ordine crescente delle misure sia secondo quello alfabetico degli elementi descrittivi. L'elaborazione grafica utilizza l'alta risoluzione derivante dalla definizione di caratteri programmabili, consentendo la visualizzazione di strisce la cui lunghezza varia in modo discreto pixel per pixel. Il grafico, inoltre, è dimensionato automaticamente e può avere uno sviluppo verticale oppure orizzontale; in entrambi i casi, se il numero di elementi supera il numero di righe e di colonne del display del TI-99/4A, essi vengono mostrati attraverso ''finestre'', alle quali si passa sequenzialmente premendo un tasto. È prevista anche la rappresentazione di valori negativi.
Dopo il RUN appare il menu principale. Premendo il numero 1 si procede all'introduzione dei dati. Questi possono rispondere o ad una funzione matematica o empirica. La funzione matematica eventualmente utilizzata va definita alla linea 1000 del programma. Le istruzioni che appaiono sul video sono comunque esaurienti.

```
100 REM DIAGRAMMI A STRISCE
110 REM ********************
120 DIM A$(100,2),X(100),I$(100)
130 CALL CHAR(128,"FF")
140 CALL CHAR(129,"0701010101010101")
150 CALL CHAR(130,"00000000FF")
160 CALL CHAR(131,"0101010101010101")
170 CALL CHAR(132,"8080808080808080")
180 CALL CHAR(136,"7F7F7F7F7F7F7F7F")
190 CALL CHAR(137,"007F7F7F7F7F7F7F")
200 CALL CHAR(138,"00007F7F7F7F7F7F")
210 CALL CHAR(139,"0000007F7F7F7F7F")
220 CALL CHAR(140,"000000007F7F7F7F")
230 CALL CHAR(141,"00000000007F7F7F")
240 CALL CHAR(142,"0000000000007F7F")
250 CALL CHAR(143,"000000000000007F")
260 CALL CHAR(144,"00FFFFFFFFFFFFFF")
270 CALL CHAR(145,"00FEFEFEFEFEFEFE")
280 CALL CHAR(146,"00FCFCFCFCFCFCFC")
290 CALL CHAR(147,"00F8F8F8F8F8F8F8")
300 CALL CHAR(148,"00F0F0F0F0F0F0F0")
310 CALL CHAR(149,"00E0E0E0E0E0E0E0")
320 CALL CHAR(150,"00C0C0C0C0C0C0C0")
330 CALL CHAR(151,"0080808080808080")
340 CALL CHAR(152,"7F7F7F7F7F7F7F")
350 CALL CHAR(153,"7F7F7F7F7F7F")
360 CALL CHAR(154,"7F7F7F7F7F")
370 CALL CHAR(155,"7F7F7F7F")
380 CALL CHAR(156,"7F7F7F")
390 CALL CHAR(157,"7F7F")
400 CALL CHAR(158,"7F")
410 CALL COLOR(14,9,1)
420 CALL COLOR(15,5,1)
430 CALL COLOR(16,9,1)
440 SP$="<SPACE>"
450 CÁLL CLEAR
460 FLAG=0
470 FLAG2=0
480 PRINT TAB(5);"DIAGRAMMI A STRISCE.":
 : : :
490 PRINT TAB(8);"1.INSERIMENTO": :
500 PRINT TAB(8);"2.MODIFICA": :
510 PRINT TAB(8);"3.TABULAZIONE": :
520 PRINT TAB(8);"4.ORDINAMENTO": :
530 PRINT TAB(8);"5.DIAGRAMMA": :
540 PRINT TAB(8);"6.FINE": : : : :
550 CALL HCHAR(4,5,128,24)
560 CALL HCHAR(7,5,128,24)
570 CALL VCHAR(4,4,131,18)
580 CALL VCHAR(4,29,132,18)
590 CALL HCHAR(22,5,128,24)
600 CALL KEY(0,K,S)
610 IF S=0 THEN 600
620 IF (K<49)+(K>54)THEN 600
630 K=K-48
640 ON K GOTO 680,1310,1810,2150,2630,44
80
650 REM *
660 REM *  INSERIMENTO
670 REM *
680 CALL CLEAR
690 PRINT TAB(7);"INSERIMENTO DATI": : :
 : :
700 INPUT "QUANTI SONO I DATI? ":N
710 PRINT
720 IF N<=100 THEN 750
730 PRINT "SPIACENTE. NON PIU` DI 100":
:
740 GOTO 700
750 PRINT : :"I DATI VARIANO SECONDO": :
760 PRINT "<1> UNA LEGGE MATEMATICA": :
770 PRINT "<2> UNA LEGGE EMPIRICA": :
780 CALL KEY(0,K,S)
790 IF S=0 THEN 780
800 IF K=49 THEN 820
810 IF K=50 THEN 1140 ELSE 780
820 CALL CLEAR
830 PRINT "PER INSERIRE LA FUNZIONE NEL"
840 PRINT "PROGRAMMA, E` NECESSARIO IN-"
850 PRINT "TERROMPERLO CON IL COMANDO"
860 PRINT "FCTN<4>=BREAK E POI SCRIVERE"
870 PRINT "LA FUNZIONE AL NUMERO DI LI-"
880 PRINT "NEA 1000 USANDO L'ISTRUZIONE"
890 PRINT "DEF. AD ESEMPIO:": : :
```

```
900 PRINT "1000 DEF F(X)=1000*(1+.12)^X"
: : :
910 PRINT "SCRITTA LA FUNZIONE, RIPREN-"
920 PRINT "DERE L'ESECUZIONE CON IL CO-"
930 PRINT "MANDO <RUN>."
940 PRINT "SE LA FUNZIONE E` GIA` STATA"
950 PRINT "SCRITTA PROSEGUIRE BATTENDO:"
: : :
960 PRINT TAB(11);"<SPACE>": :
970 CALL KEY(0,K,S)
980 IF S=0 THEN 970
990 GOSUB 4140
1000 DEF F(X)=1000*(1+.12)^X
1010 REM *
1020 REM *  FUNZIONE F(X)
1030 REM *
1040 FOR J=1 TO N
1050 IF FLAG2=1 THEN 1070
1060 A$(J,1)="F("&STR$(J)&")"
1070 A$(J,2)=STR$(F(J))
1080 IF F(J)<=MAX THEN 1100
1090 MAX=F(J)
1100 IF F(J)>=MIN THEN 1120
1110 MIN=F(J)
1120 NEXT J
1130 GOTO 1260
1140 CALL CLEAR
1150 PRINT "INSERISCI LA DESCRIZIONE E"
1160 PRINT "LA MISURA SEPARATE DA UNA":"
VIRGOLA": :
1170 FOR J=1 TO N
1180 PRINT J;
1190 INPUT A$(J,1),M
1200 A$(J,2)=STR$(M)
1210 IF M<=MAX THEN 1230
1220 MAX=M
1230 IF M>=MIN THEN 1250
1240 MIN=M
1250 NEXT J
1260 INS=1
1270 GOTO 450
1280 REM *
1290 REM *  MODIFICA
1300 REM *
1310 GOSUB 4370
1320 PRINT "MODIFICA:": :
1330 PRINT "<1> SEQUENZIALE": :
1340 PRINT "<2> SELETTIVA": :
1350 CALL KEY(0,K,S)
1360 IF S=0 THEN 1350
1370 IF K=49 THEN 1390
1380 IF K=50 THEN 1560 ELSE 1350
1390 CALL CLEAR
1400 FOR J=1 TO N
1410 PRINT J;A$(J,1);",";A$(J,2)
1420 PRINT "modifica? (y/n)"
1430 CALL KEY(0,K,S)
1440 IF S=0 THEN 1430
1450 IF K=89 THEN 1470
1460 IF K=78 THEN 1540 ELSE 1430
1470 PRINT J;
1480 INPUT A$(J,1),M
1490 A$(J,2)=STR$(M)
1500 IF M<=MAX THEN 1520
1510 MAX=M
1520 IF M>=MIN THEN 1540
1530 MIN=M
1540 NEXT J
1550 GOTO 450
1560 CALL CLEAR
1570 PRINT "QUALE DATO VUOI MODIFICARE?"
1580 INPUT D
1590 PRINT
1600 IF D<=N THEN 1630
1610 PRINT "* OUT OF RANGE!": :
1620 GOTO 1570
1630 PRINT D;A$(D,1);",";A$(D,2)
1640 PRINT "scrivi i nuovi dati":D;
1650 INPUT A$(D,1),M
1660 A$(D,2)=STR$(M)
1670 IF M<=MAX THEN 1690
1680 MAX=M
1690 IF M>=MIN THEN 1710
1700 MIN=M
1710 PRINT
1720 PRINT "ALTRE MODIFICHE? (Y/N)"
1730 CALL KEY(0,K,S)
1740 IF S=0 THEN 1730
1750 IF K=83 THEN 1570
1760 IF K=78 THEN 450 ELSE 1730
1770 GOTO 450
1780 REM *
1790 REM *  TABULAZIONE
1800 REM *
1810 GOSUB 4370
1820 FOR J=1 TO N
1830 SCREEN=0
1840 IF FLAG<>1 THEN 1880
1850 PRINT TAB(4-LEN(STR$(J)));J;"- ";A$
(X(J),1);
1860 P=20-LEN(A$(X(J),1))-LEN(A$(X(J),2)
)
1870 IF P<1 THEN 1970 ELSE 1910
1880 PRINT TAB(4-LEN(STR$(J)));J;"- ";A$
(J,1);
1890 P=20-LEN(A$(J,1))-LEN(A$(J,2))
1900 IF P<1 THEN 1950
1910 FOR V=1 TO P-1
1920 PRINT ".";
1930 NEXT V
1940 IF FLAG=1 THEN 1970
1950 PRINT TAB(28-LEN(A$(J,2)));A$(J,2)
1960 GOTO 1980
1970 PRINT TAB(28-LEN(A$(X(J),2)));A$(X(
J),2)
1980 IF J/20<>INT(J/20)THEN 2010
1990 SCREEN=20
2000 GOSUB 2050
2010 NEXT J
2020 IF SCREEN=20 THEN 450
2030 GOSUB 2050
2040 GOTO 450
2050 PRINT
2060 PRINT TAB(8);"<PREMI SPACE>"
2070 CALL KEY(0,K,S)
2080 IF S=0 THEN 2070
2090 CALL CLEAR
2100 RETURN
2110 GOTO 450
2120 REM *
2130 REM *  ORDINAMENTO
2140 REM *
2150 GOSUB 4370
```

```
2160 PRINT "SECONDO QUALE CAMPO VUOI OR-
"
2170 PRINT "DINARE LA TABELLA?": :
2180 PRINT "<1> DESCRIZIONE": :
2190 PRINT "<2> MISURA": :
2200 CALL KEY(0,K,S)
2210 IF S=0 THEN 2200
2220 IF (K<49)+(K>50)THEN 2200
2230 CALL CLEAR
2240 PRINT "UN MOMENTO... STO ELABORANDO
": : : : : : : : : :
2250 K=K-48
2260 GOSUB 2340
2270 CALL CLEAR
2280 PRINT TAB(7);"TABELLA ORDINATA": :
: :
2290 FLAG=1
2300 GOTO 1820
2310 REM *
2320 REM * ROUTINE DI ORDINAMENTO
2330 REM *
2340 FOR R=1 TO N
2350 QTA=VAL(A$(R,2))
2360 IF (K=1)+(QTA>=0)THEN 2390
2370 I$(R)=STR$(1/QTA)
2380 GOTO 2410
2390 I$(R)=A$(R,K)
2400 IF K=1 THEN 2460
2410 Z=LEN(STR$(MAX))-LEN(I$(R))
2420 IF Z=0 THEN 2460
2430 FOR W=1 TO Z
2440 I$(R)=CHR$(32)&I$(R)
2450 NEXT W
2460 NEXT R
2470 FOR J=1 TO N
2480 P=1
2490 FOR K=1 TO N
2500 IF I$(J)>I$(K)THEN 2550
2510 IF I$(J)=I$(K)THEN 2530
2520 GOTO 2560
2530 IF J>K THEN 2550
2540 GOTO 2560
2550 P=P+1
2560 NEXT K
2570 X(P)=J
2580 NEXT J
2590 RETURN
2600 REM *
2610 REM *  DIAGRAMMA
2620 REM *
2630 GOSUB 4370
2640 PRINT "DESIDERI LA RAPPRESENTAZIONE
"
2650 PRINT "DELLE MISURE": :
2660 PRINT "<1> NON ORDINATE": :
2670 PRINT "<2> ORDINATE": :
2680 CALL KEY(0,K,S)
2690 IF S=0 THEN 2680
2700 IF K=49 THEN 2745
2710 IF K<>50 THEN 2680
2720 FLAG1=1
2730 K=2
2740 GOSUB 2340
2742 GOTO 2750
2745 FOR J=1 TO N
2746 X(J)=J
```

DIAGRAMMI A STRISCE

```
2747 NEXT J
2750 IF MIN<0 THEN 3440
2760 PRINT
2770 U=MAX/19
2780 PRINT "VUOI UN DIAGRAMMA A STRISCE"
: :
2790 PRINT "<1> VERTICALI": :
2800 PRINT "<2> ORIZZONTALI": :
2810 CALL KEY(0,K,S)
2820 IF S=0 THEN 2810
2830 IF K=49 THEN 2850
2840 IF K=50 THEN 3180 ELSE 2810
2850 CALL CLEAR
2860 CALL HCHAR(21,5,128,25)
2870 CALL VCHAR(1,4,129,20)
2880 COLO=4
2890 FOR J=1 TO N
2900 IF FLAG1=1 THEN 2930
2910 Q=J
2920 GOTO 2940
2930 Q=X(J)
2940 FOR K=1 TO LEN(STR$(Q))
2950 L=ASC(SEG$(STR$(Q),K,1))
2960 CALL HCHAR(21+K,COLO+J,L)
2970 NEXT K
2980 LUNG=INT(VAL(A$(Q,2))/U)
2990 IF LUNG=0 THEN 3030
3000 FOR K=20 TO 21-LUNG STEP -1
3010 CALL HCHAR(K,COLO+J,136)
3020 NEXT K
3030 DIFF=VAL(A$(Q,2))/U-LUNG
3040 FR=INT(8*DIFF+1)
3050 IF FR=0 THEN 3070
3060 CALL HCHAR(20-LUNG,COLO+J,144-FR)
3070 IF J/25<>INT(J/25)THEN 3150
3080 CALL KEY(0,K,S)
3090 IF S=0 THEN 3080
3100 FOR H=5 TO 30
3110 CALL VCHAR(1,H,32,20)
3120 NEXT H
3130 CALL HCHAR(22,3,32,60)
3140 COLO=COLO-25
3150 NEXT J
3160 GOSUB 4040
3170 GOTO 450
3180 CALL CLEAR
3190 FOR J=1 TO N
3200 IF FLAG1=1 THEN 3230
3210 Q=J
3220 GOTO 3240
3230 Q=X(J)
3240 IF LEN(A$(Q,1))<8 THEN 3270
3250 D$=SEG$(A$(Q,1),1,7)
3260 GOTO 3280
3270 D$=A$(Q,1)
3280 PRINT TAB(8-LEN(D$));D$
3290 LUNG=INT(VAL(A$(Q,2))/U)
3300 IF LUNG=0 THEN 3320
3310 CALL HCHAR(23,11,144,LUNG)
3320 DIFF=VAL(A$(Q,2))/U-LUNG
3330 FR=INT(8*DIFF+1)
3340 IF FR=0 THEN 3360
3350 CALL HCHAR(23,11+LUNG,152-FR)
3360 IF J/20<>INT(J/20)THEN 3400
3370 IF J=N THEN 3410
3380 GOSUB 4040
3390 CALL HCHAR(24,3,32,28)
```

```
3400 NEXT J
3410 PRINT
3420 GOSUB 4040
3430 GOTO 450
3440 CALL CLEAR
3450 U=(MAX-MIN)/18
3460 Y0=INT(MAX/U)+3
3470 COLO=4
3480 CALL VCHAR(2,4,131)
3490 CALL VCHAR(3,4,129,19)
3500 CALL HCHAR(Y0,3,48)
3510 J=0
3520 GOSUB 3880
3530 FOR J=1 TO N
3540 IF FLAG1=1 THEN 3580
3550 Q=ABS(VAL(A$(J,2)))
3560 VERSO=SGN(VAL(A$(J,2)))
3570 GOTO 3600
3580 Q=ABS(VAL(A$(X(J),2)))
3590 VERSO=SGN(VAL(A$(X(J),2)))
3600 LUNG=INT(Q/U)
3610 DIFF=(Q/U-LUNG)
3620 FR=INT(8*DIFF)
3630 IF LUNG=0 THEN 3720
3640 IF VERSO<0 THEN 3690
3650 CALL VCHAR(Y0-LUNG,COLO+J,136,LUNG)
3660 IF FR=0 THEN 3720
3670 CALL VCHAR(Y0-LUNG-1,COLO+J,144-FR)
3680 GOTO 3720
3690 CALL VCHAR(Y0,COLO+J,136,LUNG)
3700 IF FR=0 THEN 3720
3710 CALL VCHAR(Y0+LUNG,COLO+J,159-FR)
3720 IF J/25<>INT(J/25)THEN 3810
3730 IF J=N THEN 3820
3740 CALL KEY(0,K,S)
3750 IF S=0 THEN 3740
3760 FOR H=5 TO 30
3770 CALL VCHAR(1,H,32,22)
3780 NEXT H
3790 COLO=COLO-25
3800 GOSUB 3880
3810 NEXT J
3820 GOSUB 4040
3830 GOTO 450
3840 6460
3850 REM *
3860 REM *  ROUTINE ESTREMI
3870 REM *
3880 FOR K=1 TO LEN(STR$(J+1))
3890 L=ASC(SEG$(STR$(J+1),K,1))
3900 CALL HCHAR(1,4+K,L)
3910 NEXT K
3920 FOR K=1 TO LEN(STR$(J+10))
3930 L=ASC(SEG$(STR$(J+10),K,1))
3940 CALL HCHAR(1,13+K,L)
3950 NEXT K
3960 FOR K=1 TO LEN(STR$(J+20))
3970 L=ASC(SEG$(STR$(J+20),K,1))
3980 CALL HCHAR(1,23+K,L)
3990 NEXT K
4000 RETURN
4010 REM *
4020 REM * SUBROUTINE SPACE
4030 REM *
4040 FOR K=1 TO 7
4050 L=ASC(SEG$(SP$,K,1))
```



```
4060 CALL HCHAR(24,12+K,L)
4070 NEXT K
4080 CALL KEY(0,K,S)
4090 IF S=0 THEN 4080
4100 RETURN
4110 REM *
4120 REM *  NUMERAZIONE F(X)
4130 REM *
4140 CALL CLEAR
4150 PRINT "VUOI NUMERARE I VALORI F(X)"
4160 PRINT "IN MODO PERSONALE? (Y/N)"
4170 CALL KEY(0,K,S)
4180 IF S=0 THEN 4170
4190 IF K=89 THEN 4210
4200 IF K=78 THEN 4260 ELSE 4170
4210 PRINT
4220 PRINT "SCRIVI IL NUMERO INIZIALE E"
4230 PRINT "L'INCREMENTO,SEPARATI DA UNA
"
4240 PRINT "VIRGOLA. ";
4250 INPUT "(N,I) ":NI,IN
4260 CALL CLEAR
4270 PRINT "UN MOMENTO... STO ELABORANDO
": : : : : : : : : :
4280 IF K=78 THEN 4330
4290 FOR J=1 TO N
4300 A$(J,1)=STR$(NI+(J-1)*IN)
4310 NEXT J
4320 FLAG2=1
4330 RETURN
4340 REM *
4350 REM * CONTROLLO INSERIMENTO
4360 REM *
4370 CALL CLEAR
4380 IF INS=1 THEN 4430
4390 PRINT "I DATI NON SONO ANCORA STATI
"
4400 PRINT "INSERITI!": : :
4410 GOSUB 4040
4420 GOTO 450
4430 CALL CLEAR
4440 RETURN
4450 REM *
4460 REM *   FINE
4470 REM *
4480 CALL CLEAR
4490 END
```

Extended ROULETTE

Chi si sente scorrere nelle vene il demone del gioco potrà con questo divertente programma realizzare i suoi sogni lasciando nel contempo intatto il suo conto in banca. Il gioco è diviso per ragioni di spazio in due blocchi: uno contiene le istruzioni relative alla grafica, l'altro è il programma vero e proprio. Caricate come al solito la prima parte e poi, seguendo le istruzioni, passate a leggere la seconda. Giocare è semplice: alla prima domanda dovete rispondere con il numero del giocatore che volete far intervenire (da 1 a 5), battere E per terminare e R per porre tutti i partecipanti alla pari. Si passa ora, manovrando il gettone blu con i tasti E,S,D,X, a decidere i vari tipi di puntata (semplice, doppia, sestine ecc.). Fissata la posizione desiderata con il tasto Q dovete ora decidere quanti dollari puntare. A questo punto ''rien ne va plus!''.

ROULETTE (I)

```
100 CALL CLEAR
110 CALL SCREEN(2)
10000 T$=RPT$("0",33):: U$=RPT$("0",48)
: V$=RPT$("0",12):: Z$=RPT$("80",14):: S
$=RPT$("01",14):: K$=RPT$("F",8)
10030 CALL CHAR(36,"FF"&Z$&"FFFF"&S$&"FF
",74,"4444444444443800385450381454380 0")
10040 CALL CHAR(40,"006E22622E286E00006E
286E22226E00006E226E22226E00006E286E2A2A
6E00",71,"0000782424382020301010101010 7C
00")
10050 CALL CHAR(44,"006E28286E4A6E00006E
2A2E62426E00006E222262426200006E2A6A2A2A
6E00",76,"0000384848483400",66,"00004444
3C040418")
10060 CALL CHAR(60,"006E2A2A6A4A6E00006E
222E62426E000642C2464446E00006828 2A6E42
6200",70,"0010107C10100000")
10070 CALL CHAR(92,"0028682A2E222200002E
62222222220000 2E68282E222E00002E6A2E2A2A
2E00")
10080 CALL CHAR(96,"001C141C14141C000024
6C2424242E00001C14141C041C00002E62222E28
2E00")
10090 CALL CHAR(100,"001C041C10101C00001
C101C04041C00001C041C04041C00001C101C141
41C00")
10100 CALL CHAR(104,T$&"0081808080 81C000
01010141C040400",108,T$&"01C040404040400
002E6A2A2A2A2E00")
10120 CALL CHAR(112,T$&"02E622622222E000
02E68282E2A2E00",116,T$&"02E6A2E22222E00
006E22226E486E00")
10140 CALL CHAR(120,T$&"06E282E62426E000
06E2A2E6A4A6E00",124,T$&"0642C6424246E00
0068286A2E226200")
10160 CALL CHAR(33,V$&K$&V$&"C0C0C0C0C0C
0C0C0",58,"0303030303030303"&K$&K$)
10210 CALL CHAR(128,"002E2A3A00003E28283
E0000382828 3E80BEA0BE8080BEA8A8BE80BEA0B
EA0BE")
10220 CALL CHAR(132,V$&"00222A3E00002E2A
3A0080A2AABE8080BE8282BE8082BEA6BE80")
10230 CALL CHAR(136,"0038282 83E000000010
1010101010000BEA0BE80BE800000C040404040C
00000")
10240 CALL CHAR(140,"0000000000 03A2A2C3E0
03E003E28283EBAACBE80BE80BEA8BE80B8A8BE8
0BEA0")
10260 CALL CHAR(88,"0103040F103F40FFFF40
3F100F04030180C020F008FC02FFFF02FC08F020
C080")
10270 CALL CHAR(84,"FFFFC3C3C3C3FFFF"&U$
)):: CALL SCREEN(4):: RUN "CS1"
```

ROULETTE (II)

```
1 OPTION BASE 1
2 CALL CLEAR :: ON BREAK NEXT :: CALL MA
GNIFY(4)
3 DIM A(10,23)
100 FOR X=1 TO 10
110 FOR Y=1 TO 23
120 READ A(X,Y)
130 NEXT Y :: NEXT X
150 DATA 1,43,4,44,7,45,10,46,13,47,16,4
8,19,49,22,50,25,51,28,52,31,53,34
160 DATA 49,50,51,52,53,54,55,56,57,58,5
9,60,61,62,63,64,65,66,67,68,69,70,71
170 DATA 2,72,5,73,8,74,11,75,14,76,17,7
7,20,78,23,79,26,80,29,81,32,82,35
180 DATA 83,84,85,86,87,88,89,90,91,92,9
3,94,95,96,97,98,99,100,101,102,103,104,
105
190 DATA 3,106,6,107,9,108,12,109,15,110
,18,111,21,112,24,113,27,114,30,115,33,1
16,36
200 DATA 117,118,119,120,121,122,123,124
,125,126,127,128,129,130,131,132,133,134
,135,136,137,138,139
210 DATA 142,142,142,200,38,38,38,200,39
,39,39,200,40,40,40,200,41,41,41,200,42,
42,42
```

```
220 DATA 140,140,140,140,140,140,140,200
,141,141,141,141,141,200,37,200,143,143,
143,200,144,144,144
230 DATA 200,200,200,200,200,200,200,200
,200,200,200,200,200,200,37,200,143,143,
143,200,144,144,144
240 DATA 145,145,145,145,145,145,145,200
,146,146,146,146,146,200,37,200,143,143,
143,200,144,144,144
1000 DISPLAY AT(2,9):"P" :: DISPLAY AT(3
,9):"12"
1010 DISPLAY AT(10,9):"D" :: DISPLAY AT(
11,9):"12"
1020 DISPLAY AT(18,9):"2" :: DISPLAY AT(
19,9):"CO"
1030 DISPLAY AT(6,9):"M" :: DISPLAY AT(7
,9):"12"
1040 DISPLAY AT(14,9):"1" :: DISPLAY AT(
15,9):"CO"
1050 DISPLAY AT(22,9):"3" :: DISPLAY AT(
23,9):"CO"
10280 FOR TT=0 TO 22 STEP 2 :: FOR SS=0-
2 TO 2 STEP 2 :: CALL HCHAR(1+TT,15+SS,3
6)
10290 CALL HCHAR(2+TT,15+SS,37):: CALL H
CHAR(1+TT,16+SS,38):: CALL HCHAR(2+TT,16
+SS,39)
10320 NEXT SS :: NEXT TT :: SS,TT=0
10330 CALL SPRITE(#1,40,2,161,113,#2,44,
2,129,113,#3,60,2,97,113,#4,92,2,65,113,
#5,96,2,33,113)
10380 CALL SPRITE(#6,100,2,1,113,#7,104,
2,1,81,#8,108,2,33,81,#9,112,2,65,81,#10
,116,2,97,81)
10430 CALL SPRITE(#11,120,2,129,81,#12,1
24,2,161,81,#13,128,2,32,148,#14,132,2,2
,148,#15,136,2,99,148)
10480 CALL SPRITE(#16,140,2,68,148,#17,8
8,2,162,148,#18,88,7,131,148)
10510 CALL HCHAR(1,11,34,2):: CALL HCHAR
(1,19,34,5):: CALL HCHAR(9,11,34,2):: CA
LL HCHAR(9,19,34,5):: CALL HCHAR(17,11,3
4,2)
10560 CALL HCHAR(17,19,34,5):: CALL HCHA
R(24,11,33,2):: CALL HCHAR(24,19,33,5)::
 CALL VCHAR(1,10,58,24):: CALL VCHAR(1,2
4,35,24)
10610 CALL HCHAR(21,19,34,5):: CALL HCHA
R(14,19,33,5)
10660 CALL HCHAR(5,11,34,2):: CALL HCHAR
(13,11,34,2):: CALL HCHAR(21,11,34,2)
10665 IP(1),IP(2),IP(3),IP(4),IP(5)=100
11990 GC=0
12000 FOR CC=1 TO 5
12010 DISPLAY AT(1+SP,23):"GHLB";CC :: D
ISPLAY AT(3+SP,23)SIZE(8):USING "#####":I
P(CC):: DISPLAY AT(3+SP,28):"K"
12020 SP=SP+4 :: NEXT CC :: SP=0
12050 IF GC=0 THEN 13020
12060 RETURN
13020 G=G+1 :: X,Y=1 :: ACCEPT AT(1,2)BE
EP SIZE(1)VALIDATE("12345RE"):GC$ :: IF
GC$="" THEN 16000 :: IF GC$="R" THEN 106
65 :: IF GC$="E" THEN 30000 ELSE GC=VAL(
GC$)
13030 CALL SPRITE(#20,84,5,X,Y)
13040 CALL KEY(0,K,ST):: IF ST=0 THEN 13
040
13060 IF K=83 THEN X=X-8 :: GOTO 13160
13070 IF K=68 THEN X=X+8 :: GOTO 13160
13080 IF K=69 THEN Y=Y-8 :: GOTO 13160
13090 IF K=88 THEN Y=Y+8 :: GOTO 13160
13100 IF K=81 THEN 13170
13110 GOTO 13040
13160 IF X<1 OR Y<1 OR X>256 OR Y>192 TH
EN X,Y=1
13162 CALL LOCATE(#20,Y,X):: GOTO 13040
13170 REM
13180 IF Y=1 THEN 13190 :: Y=(Y-1)/8+1
13190 IF X=97 THEN X=1
13200 IF X=105 THEN X=2
13210 IF X=113 THEN X=3
13220 IF X=121 THEN X=4
13230 IF X=129 THEN X=5
13240 IF X=137 OR X=89 THEN X=6
13250 IF X=81 THEN X=7
13260 IF X=145 THEN X=8
13270 IF X=153 OR X=161 THEN X=9
13280 IF X=169 THEN X=10
13290 ON WARNING NEXT :: DISPLAY AT(15,4

)SIZE(1):"K" :: ACCEPT AT(15,1)BEEP VALI
DATE(DIGIT)SIZE(3):PT(GC,G):: IF G=10 TH
EN 16000
13300 IP(GC)=IP(GC)-PT(GC,G)
13302 IF IP(GC)>=0 THEN 13310
13304 IP(GC)=IP(GC)+PT(GC,G)
13306 DISPLAY AT(15,1)SIZE(6):" NO " ::
GOTO 13290
13310 P(GC,G)=A(X,Y)
13990 GOTO 13020
15000 FOR G=1 TO G
15010 FOR GC=1 TO GC
15030 IF GC>4 THEN 15050
15040 NEXT GC
15050 NEXT G
16000 RANDOMIZE
16010 N=INT(37*RND)+1 :: IF N=37 THEN GO
SUB 20000 :: N=0 :: GOTO 16181
16015 GOSUB 23000
16020 IF N<19 THEN PS=145
16030 IF N>18 AND N<37 THEN PS=140
16040 IF N<13 THEN D=142
16050 IF N>12 AND N<25 THEN D=38
16060 IF N>24 AND N<37 THEN D=39
16070 IF I=146 THEN I$="DISPARI"
16080 IF R=143 THEN R$="ROSSO"
16090 IF R=144 THEN R$="NERO"
16130 IF I=141 THEN I$="PARI"
16170 IF PS=140 THEN PS$="PASSE"
16180 IF PS=145 THEN PS$="MANQJE"
16181 NO=N
16183 DISPLAY AT(17,1)SIZE(7):NO :: DISP
LAY AT(19,1)SIZE(7):R$ :: DISPLAY AT(20,
1)SIZE(7):I$ :: DISPLAY AT(21,1)SIZE(7):
PS$ :: IF N=0 THEN 20010
16184 FOR TT=0 TO 1000 :: NEXT TT
16190 FOR G=1 TO G
16200 FOR GC=1 TO GC
16210 IF P(GC,G)=N THEN GOSUB 17000
16220 IF P(GC,G)=R OR P(GC,G)=I OR P(GC,
G)=PS THEN GOSUB 17100
```

```
16230 IF P(GC,G)=D OR P(GC,G)=C THEN GOS
UB 17200
16240 IF P(GC,G)=S1 OR P(GC,G)=S2 THEN G
OSUB 17300
16250 IF P(GC,G)=C1 OR P(GC,G)=C2 OR P(G
C,G)=C3 OR P(GC,G)=C4 THEN GOSUB 17400
16260 IF P(GC,G)=T THEN GOSUB 17500
16270 IF P(GC,G)=V1 OR P(GC,G)=V2 OR P(G
C,G)=V3 OR P(GC,G)=V4 THEN GOSUB 17600
16280 P(GC,G)=0 :: PT(GC,G)=0
16290 IF GC>5 THEN 16310
16300 NEXT GC
16310 NEXT G
16999 GOTO 18000
17000 W=36 :: GOSUB 22000 :: RETURN
17100 W=2 :: GOSUB 22000 :: RETURN
17200 W=3 :: GOSUB 22000 :: RETURN
17300 W=6 :: GOSUB 22000 :: RETURN
17400 W=8 :: GOSUB 22000 :: RETURN
17500 W=12 :: GOSUB 22000 :: RETURN
17600 W=18 :: GOSUB 22000 :: RETURN
17800 W=.5 :: GOSUB 22000 :: RETURN
18000 GOSUB 12000 :: GOTO 20010
20000 R$,I$,PS$="" :: FOR G=1 TO G
20001 FOR GC=1 TO GC
20002 IF P(GC,G)=143 OR P(GC,G)=144 OR P
(GC,G)=146 OR P(GC,G)=141 THEN GOSUB 178
00
20003 IF P(GC,G)=140 OR P(GC,G)=145 THEN
 GOSUB 17800
20004 IF P(GC,G)=37 THEN GOSUB 17000
20005 IF GC>5 THEN 20007
20006 NEXT GC
20007 NEXT G :: GOSUB 12000 :: RETURN
20010 GC,G=0 :: GOTO 13020
21000 END
22000 PT(GC,G)=PT(GC,G)*W :: IP(GC)=IP(G
C)+PT(GC,G):: DISPLAY AT(10,1)BEEP SIZE(
7):"GHLB";GC :: DISPLAY AT(12,1)SIZE(7):
"F";PT(GC,G);K :: W=0 :: RETURN
23000 IF N=1 THEN I=146 :: R=143 :: C=40
 :: S1=118 :: C1=50 :: S2,C2,C3,C4,V3,V4
=200 :: T=117 :: V1=49 :: V2=43
23010 IF N=2 THEN I=141 :: R=144 :: C=41
 :: S1=118 :: S2,C3,C4,V4=200 :: C1=50 :
: C2=84 :: T=117 :: V1=49 :: V2=83 :: V3
=72
23020 IF N=3 THEN I=146 :: R=143 :: C=42
 :: S1=118 :: S2,C2,C3,C4,V3,V4=200 :: C
1=84 :: V1=83 :: T=117 :: V2=106
23030 IF N=4 THEN I=141 :: R=144 :: C=40
 :: S1=118 :: S2=120 :: C1=50 :: C2=52 :
: C3,C4,V4=200 :: T=119 :: V1=43 :: V2=5
1 :: V3=44
23040 IF N=5 THEN I=146 :: R=143 :: C=41
 :: S1=118 :: S2=120 :: C1=52 :: C2=86 :
: C3=50 :: C4=84 :: T=119 :: V1=51 :: V2
=72 :: V3=85 :: V4=73
23050 IF N=6 THEN I=141 :: R=144 :: C=42
 :: S1=118 :: S2=120 :: C1=86 :: C2=84 :
: C3,C4,V4=200 :: T=119 :: V1=106 :: V2=
85 :: V3=107
23060 IF N=7 THEN I=146 :: R=143 :: C=40
 :: S1=120 :: S2=122 :: C1=52 :: C2=54 :
: C3,C4,V4=200 :: T=121 :: V1=44 :: V2=5
3 :: V3=45
23070 IF N=8 THEN I=141 :: R=144 :: C=41
 :: S1=120 :: S2=122 :: C1=54 :: C2=88 :
: C3=52 :: C4=86 :: T=121 :: V1=53 :: V2
=73 :: V3=87 :: V4=74
23080 IF N=9 THEN I=146 :: R=143 :: C=42
 :: S1=120 :: S2=122 :: C1=86 :: C2=88 :
: C3,C4,V4=200 :: T=121 :: V1=107 :: V2=
87 :: V3=108
23090 IF N=10 THEN I=141 :: R=144 :: C=4
0 :: S1=122 :: S2=124 :: C1=54 :: C2=56
:: C3,C4,V4=200 :: T=123 :: V1=45 :: V2=
55 :: V3=46
23100 IF N=11 THEN I=146 :: R=144 :: C=4
1 :: S1=122 :: S2=124 :: C1=54 :: C2=88
:: C3=56 ::.C4=90 :: T=123 :: V1=55 :: V
2=74 :: V3=89 :: V4=75
23110 IF N=12 THEN I=141 :: R=143 :: C=4
2 :: S1=122 :: S2=124 :: C1=90 :: C2=88
:: C3,C4,V4=200 :: T=123 :: V1=108 :: V2
=89 :: V3=109
23120 IF N=13 THEN I=146 :: R=144 :: C=4
0 :: S1=124 :: S2=126 :: C1=56 :: C2=58
:: C3,C4,V4=200 :: T=125 :: V1=46 :: V2=
57 :: V3=47
23130 IF N=14 THEN I=141 :: R=143 :: C=4
1 :: S1=124 :: S2=126 :: C1=56 :: C2=90
:: C3=58 :: C4=92 :: T=125 :: V1=57 :: V
2=75 :: V3=91 :: V4=76
23140 IF N=15 THEN I=146 :: R=144 :: C=4
2 :: S1=124 :: S2=126 :: C1=90 :: C2=92
:: C3,C4,V4=200 :: T=125 :: V1=109 :: V2
=91 :: V3=110
23150 IF N=16 THEN I=141 :: R=143 :: C=4
0 :: S1=126 :: S2=128 :: C1=58 :: C2=60
:: C3,C4,V4=200 :: T=127 :: V1=47 :: V2=
59 :: V3=48
23160 IF N=17 THEN I=146 :: R=144 :: C=4
1 :: S1=126 :: S2=128 :: C1=58 :: C2=92
:: C3=60 :: C4=94 :: T=127 :: V1=59 :: V
2=76 :: V3,V4=77
23170 IF N=18 THEN I=141 :: R=143 :: C=4
2 :: S1=126 :: S2=128 :: C1=92 :: C2=94
:: C3,C4,V4=200 :: T=127 :: V1=110 :: V2
=93 :: V3=111
23180 IF N=19 THEN I=146 :: R=143 :: C=4
0 :: S1=128 :: S2=130 :: C1=60 :: C2=62
:: C3,C4,V4=200 :: T=129 :: V1=48 :: V2=
61 :: V3=49
23190 IF N=20 THEN I=141 :: R=144 :: C=4
1 :: S1=128 :: S2=130 :: C1=60 :: C2=94
:: C3=62 :: C4=96 :: T=129 :: V1=61 :: V
2=77 :: V3=95 :: V4=78
23200 IF N=21 THEN I=146 :: R=143 :: C=4
2 :: S1=128 :: S2=130 :: C1=94 :: C2=96
:: C3,C4,V4=200 :: T=129 :: V1=111 :: V2
=95 :: V3=112
23210 IF N=22 THEN I=141 :: R=144 :: C=4
0 :: S1=130 :: S2=132 :: C1=62 :: C2=64
:: C3=64 :: C4,V4=200 :: T=131 :: V1=49
:: V2=63 :: V3=50
23220 IF N=23 THEN I=146 :: R=143 :: C=4
1 :: S1=130 :: S2=132 :: C1=62 :: C2=96
:: C3=200 :: C4=98 :: T=131 :: V1=63 ::
V2=78 :: V3=97 :: V4=79
23230 IF N=24 THEN I=141 :: R=144 :: C=4
2 :: S1=130 :: S2=132 :: C1=96 :: C2=98
```

```
:: C3,C4,V4=200 :: T=131 :: V1=112 :: V2
=97 :: V3=113
23240 IF N=25 THEN I=146 :: R=143 :: C=4
0 :: S1=132 :: S2=134 :: C1=64 :: C2=66
:: C3=66 :: C4,V4=200 :: T=133 :: V1=50
:: V2=65 :: V3=51
23250 IF N=26 THEN I=141 :: R=144 :: C=4
1 :: S1=132 :: S2=134 :: C1=64 :: C2=98
:: C3=200 :: C4=100 :: T=133 :: V1=65 ::
 V2=79 :: V3=99 :: V4=80
23260 IF N=27 THEN I=146 :: R=143 :: C=4
2 :: S1=132 :: S2=134 :: C1=98 :: C2=100
 :: C3,C4,V4=200 :: T=133 :: V1=113 :: V
2=99 :: V3=114
23270 IF N=28 THEN I=141 :: R=144 :: C=4
0 :: S1=134 :: S2=136 :: C1=66 :: C2=68
:: C3=68 :: C4,V4=200 :: T=135 :: V1=51
:: V2=67 :: V3=52
23280 IF N=29 THEN I=146 :: R=144 :: C=4
1 :: S1=134 :: S2=136 :: C1=66 :: C2=100
 :: C3=200 :: C4=102 :: T=135 :: V1=67 :
: V2=80 :: V3=101 :: V4=81
23290 IF N=30 THEN I=141 :: R=143 :: C=4
2 :: S1=134 :: S2=136 :: C1=100 :: C2=10
2 :: C3,C4,V4=200 :: T=135 :: V1=114 ::
V2=101 :: V3=115
23300 IF N=31 THEN I=146 :: R=144 :: C=4
0 :: S1=136 :: S2=138 :: C1=68 :: C2=70
:: C3=70 :: C4,V4=200 :: T=137 :: V1=52
:: V2=69 :: V3=53
23310 IF N=32 THEN I=141 :: R=143 :: C=4
1 :: S1=136 :: S2=138 :: C1=68 :: C2=102
 :: C3=200 :: C4=104 :: T=137 :: V1=69 :
: V2=81 :: V3=103 :: V4=82
23320 IF N=33 THEN I=146 :: R=144 :: C=4
2 :: S1=136 :: S2=138 :: C1=102 :: C2=10
4 :: C3,C4,V4=200 :: T=137 :: V1=115 ::
V2=103 :: V3=116
23330 IF N=34 THEN I=141 :: R=143 :: C=4
0 :: S1=138 :: S2,C2,V3,V4,C4=200 :: C1=
70 :: C3=62 :: T=139 :: V1=53 :: V2=71
23340 IF N=35 THEN I=146 :: R=144 :: C=4
1 :: S1=138 :: S2,C4,V4=200 :: C1=70 ::
C2=104 :: C3=62 :: T=139 :: V1=71 :: V2=
82 :: V3=105
23350 IF N=36 THEN I=141 :: R=143 :: C=4
2 :: S1=138 :: S2,C2,C4,V3,V4=200 :: C1=
104 :: C3=62 :: T=139 :: V1=116 :: V2=10
5
23360 RETURN

30000 CALL CLEAR :: END
```

Questo mese il ''**TI 99 NEWSOFT**'' compie un anno.
Quale modo migliore di augurargli lunga vita
se non quello di abbonarsi?
Con sole L. 95.000 riceverete
11 numeri della rivista e in più avrete diritto
agli sconti sul software e l'hardware offerto da

COMPUMARKET

Per abbonarsi basta inviare
un assegno non trasferibile o un vaglia postale a
NEWSOFT snc - Via S. Jacini, 4 - 20121 Milano

Ti Basic

IL GIOCO DEL 15

Vi proponiamo il veterano gioco del 15, la classica tavoletta con i numeri mobili che tutti noi abbiamo avuto tra le mani almeno una volta, magari durante un noioso viaggio in treno o in una notte d'insonnia. Usando E,S,D,X, e con il tasto ALPHA LOCK inserito, possiamo spostare, uno alla volta, i quattordici numeri disposti casualmente nelle quindici caselle del quadrato, ed ogni vostra mossa verrà meticolosamente segnalata sul video, in modo che, se lo desiderate, potete organizzare dei veri e propri tornei di 15 direttamente a casa vostra.

```
1 CALL CLEAR
3 PRINT "      IL GIOCO DEL 15": : : : :
: : : :
5 PRINT "          E                    us
a:  S D                              X
                                    "
7 PRINT "     inserire  -ALPHA LOCK-"
9 FOR T=1 TO 900
11 NEXT T
13 CALL CLEAR
17 CALL CHAR(46,"0000000000FFFFFF")
19 CALL CHAR(42,"0000000000FFFFFF")
21 CALL CHAR(44,"C0C0C0C0C0C0C0C0")
23 CALL CHAR(45,"")
25 CALL CHAR(47,"C0C0C0C0C0FFFFFF")
27 CALL CHAR(58,"002E6A2A2A2A2E00")
29 CALL CHAR(59,"00246C2424242400")
31 CALL CHAR(60,"002E6A222E282E00")
33 CALL CHAR(61,"002E622622222E00")
35 CALL CHAR(62,"002868282A2E2200")
37 CALL CHAR(63,"002E682E22222E00")
39 M=-1
41 CALL HCHAR(1,1,45,1200)
43 Y=65
45 FOR T=1 TO 12 STEP +3
47 Y=Y-1
49 FOR H=1 TO 12 STEP +3
51 IF H=1 THEN 55 ELSE 53
53 Y=Y-1
55 CALL HCHAR(4+H,9+T,Y+F)
57 CALL SOUND(30,-1,1)
59 NEXT H
61 NEXT T
63 CALL HCHAR(5,10,127)
```

```
65 CALL HCHAR(3,9,42,12)
67 CALL HCHAR(6,9,46,13)
69 CALL HCHAR(9,9,46,13)
71 CALL HCHAR(12,9,46,13)
73 CALL HCHAR(15,9,42,12)
75 CALL VCHAR(4,9,44,12)
77 CALL VCHAR(6,9,47)
79 CALL VCHAR(9,9,47)
81 CALL VCHAR(12,9,47)
83 CALL VCHAR(15,9,47)
85 CALL VCHAR(4,12,44,11)
87 CALL VCHAR(6,12,47)
89 CALL VCHAR(9,12,47)
91 CALL VCHAR(12,12,47)
93 CALL VCHAR(15,12,47)
95 CALL VCHAR(4,15,44,11)
97 CALL VCHAR(6,15,47)
99 CALL VCHAR(9,15,47)
101 CALL VCHAR(12,15,47)
103 CALL VCHAR(15,15,47)
105 CALL VCHAR(4,18,44,12)
107 CALL VCHAR(6,18,47)
109 CALL VCHAR(9,18,47)
111 CALL VCHAR(12,18,47)
113 CALL VCHAR(15,18,47)
115 CALL VCHAR(4,21,44,12)
119 ME$="MOSSA n "
121 FOR A=1 TO 8
123 CALL HCHAR(22,2+A,ASC(SEG$(ME$,A,1))
)
125 NEXT A
127 FOR T=2 TO 14 STEP +3
129 FOR H=10 TO 21 STEP +3
131 CALL GCHAR(T,H,L)
133 IF L=127 THEN 139
135 NEXT H
137 NEXT T
139 CALL SOUND(5,470,1)
141 M=M+1
143 M$=STR$(M)
145 FOR P=1 TO LEN(M$)
147 CALL HCHAR(22,10+P,ASC(SEG$(M$,P,1))
)
149 NEXT P
151 CALL KEY(0,K,S)
153 CALL HCHAR(20,4,126,G)
155 IF S=0 THEN 151
157 IF K=69 THEN 167
159 IF K=83 THEN 177
```

```
161 IF K=88 THEN 197
163 IF K=68 THEN 187
165 GOTO 151
167 CALL GCHAR(T+3,H,L;
169 IF (L=45)+(L=44)+(L=47)THEN 151
171 CALL HCHAR(T,H,L)
173 CALL HCHAR(T+3,H,127)
175 GOTO 115
177 CALL GCHAR(T,H+3,L)
179 IF (L=44)+(L=45)+(L=47)THEN 151
181 CALL HCHAR(T,H,L)
183 CALL HCHAR(T,H+3,127)
185 GOTO 115
187 CALL GCHAR(T,H-3,L)
189 IF (L=45)+(L=44)+(L=47)THEN 151
191 CALL HCHAR(T,H,L)
193 CALL HCHAR(T,H-3,127)
195 GOTO 115
197 CALL GCHAR(T-3,H,L)
199 IF (L=45)+(L=44)+(L=45)THEN 151
201 CALL HCHAR(T,H,L)
203 CALL HCHAR(T-3,H,127)
205 GOTO 115
```


OFFERTA SPECIALE

è pronto finalmente il manuale per l'extended basic in italiano!

ORDINATELO CON LA CARTOLINA DEL COMPUMARKET

L. 20.000 (abbonati L. 18.000)

Extended CRAZY KOALA

In questo videogame impersonate un piccolo ma tenace koala, che vuole a tutti i costi raggiungere la sua compagna in cima ad una pericolosa impalcatura. Dall'alto un koala rivale cercherà di impedirvi l'ascesa, facendo cadere pesanti massi blu. A voi il compito di evitarli, cercando anche di cogliere lungo il percorso quanta più frutta possibile. Oltre a dover evitare i colpi dell'avversario in alto, ogni tanto un grosso corvaccio giallo interromperà il vostro cammino, tentando di beccarvi sul collo. Non vi rimane che scappare, salendo o scendendo le scale il più in fretta possibile, o addirittura saltando.
Il gioco è a tempo, allo scadere del quale perdete punti e koala. Per ogni koala raggiunta guadagnate punti e passate immediatamente ad un nuovo schermo, sempre più veloce e pericoloso.
Il movimento del koala avviene premendo i tasti E,S,D,X (maiuscoli), secondo la direzione delle frecce. Per saltare usate J, tenendo presente che il salto dura qualche frazione di secondo, durante la quale non siete in grado di correggere il movimento del vostro piccolo eroe.

```
100 REM *****CRAZY  COALA***
110 CALL CLEAR :: GOSUB 1510
120 A1$="000006090979DEFEFDFB371F1F1D183
0000000000000000F0F8FCFEFFFF1E7C" :: CA
LL CHAR(96,A1$)
130 A2$="00000C12127EDCFEFF7F3F1F1F1F3B6
0000000000000000F0F8FCFCFEFEE78101" :: CA
LL CHAR(100,A2$)
140 A3$="0000000000000000000F1F3F7FFFFF783
E0000609090909E7B7FBFDEECF8F8B8180C" :: CA
LL CHAR(104,A3$)
150 A4$="0000000000000000071F3F3F7F7FE7818
000003048487E3B7FFFFEFCF8F8F8DC06" :: CA
LL CHAR(108,A4$)
160 A5$="0C1F1F0F0723373F1F0F27373F1F0F0
030F8F8F0E0C4ECFCF8F0E4ECFCF8F" :: CALL
CHAR(112,A5$)
170 A6$="033C767F3E4DEBF77F3F07030000000
0804080000080C0E0F8F8F8F838180C04" :: CA
LL CHAR(116,A6$)
180 A7$="0102010000010307 1F1F1F1F1C18302
0C03C6EFE7CB2D7EFFEFCE0C" :: CALL CHAR(1
20,A7$):: A8$="1028686C0B333000000000000
0000000000000000000000000" :: CALL CHAR
(72,A8$)
190 A9$="0402020606 0E1C700000000000000000
0000000000000000000" :: CALL CHAR(76,A9$
)
200 A10$="10087EFFFFFF7E3C0000000000000000
0000000000000000000" :: CALL CHAR(80,A1
0$)
```

```
210 B1$="04083E5D3E3E3E1C0000000000000000
0000000000000000000" :: CALL CHAR(84,B1$
)
220 B2$="0810103C7E7E7E3C0000000000000000
0000000000000000000" :: CALL CHAR(88,B2
$)
230 B3$="08102844E4EE4E040000000000000000
0000000000000000000" :: CALL CHAR(92,B3
$)
240 B4$="1010387C387C7C380000000000000000
0000000000000000000" :: CALL CHAR(68,B4
$)
250 B5$="0810383387C7C7C380000000000000000
0000000000000000000" :: CALL CHAR(140,B
5$)
260 B6$="0000000000000000003B6FFFFF770000
0000000000000031FFEF0FFFFFF" :: CALL CH
AR(124,B6$)
270 B7$="0000000000000000003B6FFFFF770000
0000000060E1E3E7CF8F1FFFFFFF" :: CALL CH
AR(128,B7$)
280 B8$="0006060305070300D1E3F37373636362
6006060C0A0E0C0B078FCECEC6C6C6C64" :: CA
LL CHAR(132,B9$)
290 B9$="0000060909090D02070D1E3F3F3F3F3F1
D0000182424ECD0F0F0D030F0D0D0D0D0" :: CA
LL CHAR(136,B9$)
300 B10$="000000000030707F1F3F3F3F3F3F1F1F
0F00000000E0F0F8F8FCFCFCFCFCF8F0E0" :: C
ALL CHAR(36,B10$)
310 CALL CHAR(40,"FF8080FFFF8080FF"):: C
ALL CHAR(41,"FF0101FFFF0101FF"):: CALL C
HAR(45,"FFFEFFFEFFFEFFFE"):: CALL CHAR(4
6,"7FFF7FFF7FFF7FFF")
320 CALL CHAR(42,"80C0FFFFFFFFFF80"):: C
ALL CHAR(43,"0103FFFFFFFFFF01"):: CALL C
HAR(44,"0000FFFFFFFFFFFF"):: CALL CHAR(3
4,"FCFEFFFFFFFFFEFC")
330 CALL CHAR(35,"3F7FFFFFFFF7F3F"):: C
ALL CHAR(58,"00000000000000FF"):: CALL C
HAR(59,"00000000000000FF"):: CALL CHAR(6
0,"00000000000000FF")
340 CALL CHAR(61,"00000000FF"):: CALL CH
AR(62,"000000FF"):: CALL CHAR(63,"0000FF
"):: CALL CHAR(64,"00FF"):: CALL CHAR(65
,"FF"):: CALL CLEAR
350 CALL MAGNIFY(3):: CALL CLEAR :: CALL
 SCREEN(1):: FOR DRI=1 TO 14 :: CALL COL
OR(DRI,6,1):: NEXT DRI
```

```
360 FOR A=1 TO 15 STEP 2 :: FOR F=1 TO 2
 :: CALL HCHAR(23,1+(A+F),57+(A+1)/2)::
NEXT F :: NEXT A
370 FOR A=1 TO 11 STEP 2 :: FOR F=1 TO 2
 :: CALL HCHAR(22,17+(A+F),57+(A+1)/2)::
 NEXT F :: NEXT A
380 FOR A=1 TO 15 STEP 2 :: FOR F=1 TO 2
 :: CALL HCHAR(14,1+(A+F),57+(A+1)/2)::
NEXT F :: NEXT A
390 FOR A=1 TO 11 STEP 2 :: FOR F=1 TO 2
 :: CALL HCHAR(13,17+(A+F),57+(A+1)/2)::
 NEXT F :: NEXT A
400 FOR A=1 TO 15 STEP 2 :: FOR F=1 TO 2
 :: CALL HCHAR(6,1+(A+F),57+(A+1)/2):: N
EXT F :: NEXT A
410 FOR A=1 TO 11 STEP 2 :: FOR F=1 TO 2
 :: CALL HCHAR(5,17+(A+F),57+(A+1)/2)::
NEXT F :: NEXT A
420 FOR A=1 TO 15 STEP 2 :: FOR F=1 TO 2
 :: CALL HCHAR(18,1+(A+F),66-(A+1)/2)::
NEXT F :: NEXT A
430 FOR A=1 TO 11 STEP 2 :: FOR F=1 TO 2
 :: CALL HCHAR(19,17+(A+F),66-(A+1)/2)::
 NEXT F :: NEXT A
440 FOR A=1 TO 15 STEP 2 :: FOR F=1 TO 2
 :: CALL HCHAR(10,1+(A+F),66-(A+1)/2)::
NEXT F :: NEXT A
450 CALL HCHAR(3,25,65,5)
460 FOR A=1 TO 11 STEP 2 :: FOR F=1 TO 2
 :: CALL HCHAR(11,17+(A+F),66-(A+1)/2)::
 NEXT F :: NEXT A
470 CALL VCHAR(1,1,46,24):: CALL VCHAR(1
,2,45,24):: CALL VCHAR(1,31,46,24):: CAL
L VCHAR(1,32,45,24)
480 CALL VCHAR(19,17,40,4):: CALL VCHAR(
19,18,41,4):: CALL VCHAR(15,3,40,3):: CA
LL VCHAR(15,4,41,3):: CALL VCHAR(11,21,4
0,3)
490 CALL VCHAR(11,22,41,3):: CALL VCHAR(
7,5,40,3):: CALL VCHAR(7,6,41,3):: CALL
VCHAR(3,23,40,3):: CALL VCHAR(3,24,41,3)
500 CALL SPRITE(#1,92,7,150,40):: CALL S
PRITE(#2,72,8,152,120):: CALL SPRITE(#3,
76,11,110,96):: CALL SPRITE(#4,80,9,120,
232)
510 CALL SPRITE(#5,88,12,82,32):: CALL S
PRITE(#6,88,10,82,128):: CALL SPRITE(#7,
88,4,50,88):: CALL SPRITE(#8,140,11,56,2
32)
```

```
520 CALL SPRITE(#9,68,4,17,40):: CALL SP
RITE(#10,84,10,17,112):: CALL SPRITE(#11
,132,11,1,120):: CALL SPRITE(#12,136,16,
1,216)
530 CALL SPRITE(#13,104,16,168,17):: RAN
DOMIZE :: IF XZ=0 THEN SP=15 :: L=1 :: X
=166 :: Y=17 :: PL=2 :: PR=-2 :: TF=2 ::
 E=1 :: G=1 :: BONUS=2500 :: TIM=0 :: SC
ORE=0 :: KOALA=3
540 DISPLAY AT(24,1)SIZE(5):SCORE :: DIS
PLAY AT(24,10)SIZE(5):BONUS :: DISPLAY A
T(24,20)SIZE(2):KOALA
550 CALL SPRITE(#15,36,14,1,14,SP,0)
560 CALL KEY(0,K,S):: IF S<>0 THEN GOSUB
 690
570 CALL COINC(#13,#15,13,C):: IF C=-1 T
HEN 830
580 CALL POSITION(#15,M,N):: IF M>164 TH
EN GOSUB 670
590 CALL KEY(0,K,S):: IF S<>0 THEN GOSUB
 690
600 CALL COINC(#13,#15,13,C):: IF C=-1 T
HEN 830
610 IF T=0 THEN A=INT(RND*5):: IF A=1 TH
EN 750
620 IF T=1 THEN CALL POSITION(#14,O,P)::
 IF P<20 THEN CALL DELSPRITE(#14):: T=0
ELSE 630
630 IF T=1 THEN CALL COINC(#13,#14,10,C)
:: IF C=-1 THEN 830
640 IF T=1 THEN TF=-TF :: CALL PATTERN(#
14,126+TF)
650 B=B+1 :: IF B=7 THEN B=1 :: BONUS=BO
NUS-100 :: DISPLAY AT(24,10)SIZE(5):BONU
S :: IF BONUS=0 THEN 830 ELSE 560
660 GOTO 560
670 Q=INT(RND*200)+1 :: CALL LOCATE(#11,
1,Q):: CALL LOCATE(#15,14,Q):: CALL KEY(
0,K,S):: IF S<>0 THEN GOSUB 690
680 RETURN
690 IF K=68 THEN 910
700 IF K=83 THEN 960
710 IF K=69 THEN 1010
720 IF K=74 THEN 1220
730 IF K=88 THEN 1140
740 RETURN
750 H=INT(RND*5)+1 :: T=1 :: IF H/2=INT(
H/2)THEN I=1 ELSE I=-1
760 ON H GOTO 770,780,790,800,810
770 LD=20 :: GOTO 820
```

```
780 LD=67 :: GOTO 820
790 LD=83 :: GOTO 820
800 LD=134 :: GOTO 820
810 LD=154 :: GOTO 820
820 CALL SPRITE(#14,128,11,LD,240,I,-12)
:: GOTO 620
830 CALL MOTION(#13,0,0,#15,0,0):: IF T=
1 THEN CALL MOTION(#14,0,0)
840 KOALA=KOALA-1 :: DISPLAY AT(24,20)SI
ZE(2):KOALA :: C$="0000000000000F180F000
00F3F7FFFFF0000000000000F018F00000F0FCFEF
FFF" :: CALL CHAR(104,C$):: CALL PATTERN
(#13,104)
850 CALL SOUND(750,175,1,349,1):: CALL S
OUND(700,220,1,440,1):: CALL SOUND(300,2
08,1,415,1):: CALL SOUND(300,247,1,494,1
)
860 CALL SOUND(800,175,1,349,1):: CALL D
ELSPRITE(#13,#14,#15):: B=1 :: XZ=1 :: X
=168 :: Y=17 :: BONUS=2500 :: L=1 :: G=1
 :: E=1 :: DISPLAY AT(24,10)SIZE(5):BONU
S
870 CALL CHAR(104,A3$):: IF KOALA=0 THEN
 1390
880 CALL SPRITE(#13,104,16,X,Y):: GOTO 5
50
890 CALL MOTION(#13,10,0)
900 CALL POSITION(#13,KO,KI):: IF KO>172
 THEN 830 ELSE 900
910 PR=-PR :: CALL PATTERN(#13,106+PR)::
 G=G+1 :: IF G=3 THEN G=1
920 IF G<>1 THEN 940
930 IF L/2=INT(L/2)THEN F=1 ELSE F=-1
940 X=X+F :: F=0 :: Y=Y+8 :: DIR=1 :: IF
 Y>240 THEN 890
950 CALL LOCATE(#13,X,Y):: RETURN
960 PL=-PL :: CALL PATTERN(#13,98+PL)::
G=G-1 :: IF G=0 THEN G=2
970 IF G<>2 THEN 990
980 IF L/2=INT(L/2)THEN F=-1 ELSE F=1
990 X=X+F :: F=0 :: Y=Y-8 :: DIR=-1 :: I
F Y<=7 THEN 890
1000 CALL LOCATE(#13,X,Y):: RETURN
1010 NB=1 :: ON L GOTO 1090,1100,1110,11
20,1130
1020 IF Y<>LL THEN RETURN
1030 CALL PATTERN(#13,112):: CALL MOTION
(#13,-8,0)
1040 CALL POSITION(#13,V,W):: IF V<=LT-1
```

```
2 THEN X=X-(LB-LT):: CALL MOTION(#13,0,0
):: CALL LOCATE(#13,X,Y):: CALL PATTERN(
#13,96):: L=L+1 ELSE 1060
1050 IF L=6 THEN 1420 ELSE RETURN
1060 CALL COINC(#13,#15,13,C):: IF C=-1
THEN 830
1070 IF T=1 THEN CALL COINC(#13,#14,10,C
):: IF C=-1 THEN 830
1080 GOTO 1040
1090 LB=178 :: LL=129 :: LT=145 :: IF NB
=1 THEN 1020 ELSE 1160
1100 LB=137 :: LL=17 :: LT=112 :: IF NB=
1 THEN 1020 ELSE 1160
1110 LB=102 :: LL=161 :: LT=80 :: IF NB=
1 THEN 1020 ELSE 1160
1120 LB=76 :: LL=33 :: LT=50 :: IF NB=1
THEN 1020 ELSE 1160
1130 LB=37 :: LL=177 :: LT=16 :: IF NB=1
 THEN 1020 ELSE 1160
1140 IF L=1 THEN RETURN
1150 NB=0 :: ON L-1 GOTO 1090,1110,1120,
1130
1160 IF Y<>LL THEN RETURN
1170 CALL PATTERN(#13,112):: CALL MOTION
(#13,8,0)
1180 CALL POSITION(#13,V,W):: IF V=LB-16
 THEN X=X+(LB-LT):: CALL MOTION(#13,0,0)
:: CALL LOCATE(#13,X,Y):: CALL PATTERN(#
13,104):: L=L-1 :: RETURN
1190 CALL COINC(#13,#15,13,C):: IF C=-1
THEN 830
1200 IF T=1 THEN CALL COINC(#13,#14,12,C
):: IF C=-1 THEN 830
1210 GOTO 1180
1220 SP1=L*2-1 :: SP2=L*2 :: IF DIR=-1 T
HEN CALL PATTERN(#13,116)ELSE CALL PATTE
RN(#13,120)
1230 CALL MOTION(#13,-8,0)
1240 IF T=1 THEN CALL COINC(#13,#14,12,C
):: IF C=-1 THEN 830
1250 CALL COINC(#13,#15,13,C):: IF C=-1
THEN 830
1260 CALL COINC(#13,#SP1,8,CO):: IF CO=-
1 THEN 1300
1270 CALL COINC(#13,#SP2,8,COI):: IF COI
=-1 THEN 1300
1280 CALL POSITION(#13,V,W):: IF V<=X-20
 THEN 1340
1290 GOTO 1240
```

```
1300 IF CO=-1 THEN CALL DELSPRITE(#SP1):
: JR=SCORE :: SCORE=SCORE+SP1*50 ELSE CA
LL DELSPRITE(#SP2):: JR=SCORE :: SCORE=S
CORE+SP2*50
1310 CALL SOUND(200,1100,1):: IF JR<1000
0 AND SCORE>10000 THEN SCORE=SCORE-10000
 :: TIM=TIM+10000
1320 IF HB=1 THEN RETURN ELSE HB=0
1330 DISPLAY AT(24,1)SIZE(5):SCORE
1340 CALL MOTION(#13,8,0)
1350 IF T=1 THEN CALL COINC(#13,#14,12,C
):: IF C=-1 THEN 830
1360 CALL COINC(#13,#15,15,C):: IF C=-1
THEN 830
1370 CALL POSITION(#13,V,W):: IF V<X-3 T
HEN 1350 ELSE CALL MOTION(#13,0,0):: CAL
L LOCATE(#13,X,Y)
1380 IF DIR=1 THEN CALL PATTERN(#13,104)
:: RETURN ELSE CALL PATTERN(#13,96):: RE
TURN
1390 CALL CLEAR :: CALL CHARSET :: FOR C
OL=1 TO 8 :: CALL COLOR(COL,16,1):: NEXT
 COL :: CALL SCREEN(5)
1400 CALL DELSPRITE(ALL):: XZ=0 :: PRINT
 "PUNTEGGIO:";SCORE+TIM:"":"":"":"PREMI
< ENTER > PER GIOCARE":"":"":"":""
1410 INPUT ANS$ :: GOTO 120
1420 CALL KEY(0,K,S):: IF K=68 THEN GOSU
B 910 ELSE 1420
1430 CALL DISTANCE(#13,#12,DIST):: IF DI
ST>500 THEN 1420
1440 FOR S1=1 TO 27 :: READ D,E :: CALL
SOUND(D,E,0):: NEXT S1
1450 SCORE=SCORE+BONUS :: DISPLAY AT(24,
1)SIZE(5):SCORE
1460 DATA 150,294,150,330,150,349,150,39
2,150,440,150,440,300,440,150,523,150,40
6,150,440,150,392
1470 DATA 300,440,300,440,150,466,150,39
2,150,392,150,392,150,440,150,349,150,14
9,150,349,150,440
1480 DATA 150,392,150,349,150,330,300,29
4,300,294
1490 RESTORE 1460 :: XZ=1 :: Y=17 :: X=1
68 :: BONUS=2500 :: L=1 :: G=1 :: E=1 ::
 IF SP>30 THEN 500 ELSE SP=SP+2
1500 GOTO 500
1510 DISPLAY AT(10,9):"CRAZY KOALA":"":T
AB(11);"(@)1984"
1520 RETURN
```

NUMERO DOPPIO

TI 99 NEWSOFT

L. 10.000

Numero 11 - anno II° - luglio-agosto 1985

LA SECONDA PUNTATA DI: COME COSTRUIRE UN ADVENTURE

TANTI PROGRAMMI PER LA VOSTRA ESTATE

TI99IUC.it

IN SOFTWARE & SOFTWARE
Algol - Ti Water Skiing
Tilandia - Politicanza - Life

SOMMARIO

☐ **TI BASIC**
Impariamo a risparmiare velocità pag. 3

☐ **PROGRAMMAZIONE**
L'avventura continua pag. 4

☐ **TI 99/4A**
Le memorie di massa pag. 7

☐ **MINI MEMORY**
Alta risoluzione pag. 8

☐ **DIDATTICA**
Supercalc pag. 10

☐ **TI WATER SKIING**
Un programma molto, molto estivo pag. 12

☐ **BIBLIOTECA**
Un modo pratico per ordinare i vostri libri pag. 13

☐ **TI-PUFFI**
Un'avventura in Pufflandia pag. 14

☐ **ALGOL**
La versione del noto gioco per TI 99 + Extended pag. 15

☐ **LIFE**
Il celebre algoritmo di J.H. Conway pag. 18

☐ **POLITICANZA**
Tutto ciò che volevate sapere sul linguaggio dei politici pag. 21

☐ **TILANDIA**
Il Re Astolfo ha bisogno della vostra collaborazione pag. 22

☐ **RECENSIONI**
Fun-Pac e Robopods, due cassette della Virgin pag. 24

☐ **LA VIGNETTA** pag. 25

☐ **PICCOLI ANNUNCI** pag. 26

☐ **COMPIUMARKET** pag. 29

TI 99 NEWSOFT

ANNO II° - Numero UNDICI
LUGLIO-AGOSTO 1985
MENSILE
Direttore responsabile:
ANTONIO C. LOSITO
Coordinatore Editoriale:
VINICIO CIVININI
Direttore Amministrativo:
ENRICO CALLERIO
Art Director:
BEPPE RE-FRASCHINI
Illustrazioni:
MARIA TERESA COCOZZA
Segreteria:
CRISTINA FRISON
Redazione:
FIORENZA AURIEMMA
Collaboratori:
SERGIO BORSANI - GIANPAOLO BOTTIN - DANIELE CATALFAMO - PAOLO CIVARDI - STEFANO DELLI PONTI - FRANCESCO MACALI - MAURO MATTIAZZI - EZIO MONTINI - LEVIO PEROTTI - RICCARDO ROSSI - MARCO SQUINTANI - ALBERTO STRAFILE - SEBASTIANO TOMASELLO
TI 99 NEWSOFT è una pubblicazione della **NEWSOFT snc.**
Via Stefano Jacini, 4 - 20121 MI
Tel. 02/807464 - Telex 324284
REGISTRAZIONE AL TRIBUNALE DI MILANO N. 323 - DEL 7.7.84

© COPYRIGHT NEWSOFT snc
TUTTI I DIRITTI RISERVATI
Manoscritti e foto originali, anche se non pubblicati, non si restituiscono ed è vietata la riproduzione anche parziale di testi, fotografie e listati.

Downloaded from
www.ti99iuc.it

Abbonamenti: l'abbonamento annuo a 11 numeri di **TI 99 NEWSOFT** costa L. 95.000 per l'Italia; L. 160.000 per l'Europa (spedizione aerea); L. 200.000 per tutti gli altri paesi. L'importo, per assegno non trasferibile o per vaglia postale, va inviato a **NEWSOFT snc,** Via S. Jacini, 4 - 20121 Milano
Stampa: LALITOTIPO srl - Milano
Concessionaria per la distrib.:
EUROSTAMPA Srl - C.so Vittorio Emanuele, 11 - 10123 Torino
Telefono 011/538166-7

L'editore non si assume responsabilità alcuna sul contenuto degli articoli e/o software inviati. Per gli articoli firmati e/o siglati da collaboratori esterni, la redazione si assume la responsabilità prevista dalle leggi sulla stampa.

Tutti possono collaborare a TI 99 NEWSOFT, sia con articoli che con programmi. Scriveteci o telefonateci in redazione per avere tutti i ragguagli.

RISPARMIARE VELOCITÀ

di MARCO SQUINTANI

Il TI BASIC, come più volte ricordato, pur essendo un linguaggio molto potente e versatile, ha tuttavia la profonda lacuna della scarsa velocità di esecuzione dei comandi.
Cosa è dunque possibile fare per migliorare la situazione? Molti accorgimenti che si possono usare sono gli stessi prescritti nella precedente puntata, a proposito del risparmio di memoria. Ciò significa, in molti casi, poter aumentare l'efficienza del vostro programma, sia per quanto riguarda la velocità che le dimensioni. Mossa fondamentale è l'eliminazione fisica delle righe di commento (REM), dato che il BASIC, al loro incontro, le ignora con conseguente perdita di tempo.
Usare le variabili al posto delle costanti è poi, forse, il consiglio più importante per aumentare la velocità d'esecuzione del vostro listato. Infatti, ci vuole più tempo a convertire una costante nella sua rappresentazione in virgola mobile che a prendere il valore di una variabile semplice o di un elemento di una matrice, con un particolare riguardo nel caso ci si trovi all'interno di cicli FOR... NEXT o altre parti iterative.
Ancora sulle variabili. Quelle che vengono incontrate per prime durante l'esecuzione di un programma BASIC vengono messe all'inizio della tabella delle variabili. Per esempio, nello statement del tipo 10 A=O, 20 B=A, 30 C=1, il computer assegnerà ad A il primo, a B il secondo e a C il terzo posto nella tabella dei simboli. Più avanti nel programma, quando si incontra un riferimento alla variabile A, il BASIC dovrà esaminare un solo elemento della tabella dei simboli per trovare A, due per trovare B, tre per trovare C, ecc. ecc. Limitate al massimo l'uso ripetuto di istruzioni per il salto incondizionato (GOTO, GOSUB... RETURN).
Nella lettura dei caratteri da schermo, infine, usate le CALL GCHAR solo quando ce ne sia disperato ed assoluto bisogno. Aggirate l'ostacolo ponendo confronti di posizione nelle IF... THEN... ELSE.
Per esempio: siano A e B le coordinate del carattere del vostro omino che, capitato nella posizione 15,10 della mina deve, purtroppo, soccombere. Ponete allora la questione in questi termini, digitando al posto di CALL GCHAR (A,B,Z,) e IF Z = CH THEN XXX (dove si intende per Z una variabile di comodo, per CH il codice carattere della mina e per XXX il numero di linea dove avviene l'eliminazione dell'omino), l'istruzione IF (A=15) * (B=10) THEN XXX, otterrete ottimi e confortanti risultati.
Non lasciatevi abbattere troppo dalla triste sorte del vostro eroe; sursum corda (in alto i cuori) perché nel prossimo numero affronteremo un'altro affascinante tema.
Non mancate. ■

ADVENTURLAND L'AVVENTURA CONTINUA

di DANIELE CATALFAMO

Eccoci di nuovo qua, e, come promesso, prima di continuare la nostra discussione sulla costruzione di un adventure-game, eccovi la soluzione di quello presentatovi il mese scorso; in figura 1 potete osservare la mappa dell'adventure, le stanze tratteggiate sono quelle in cui bisogna entrare usando degli stratagemmi.
Questa è la sequenza di comandi per risolvere l'avventura:

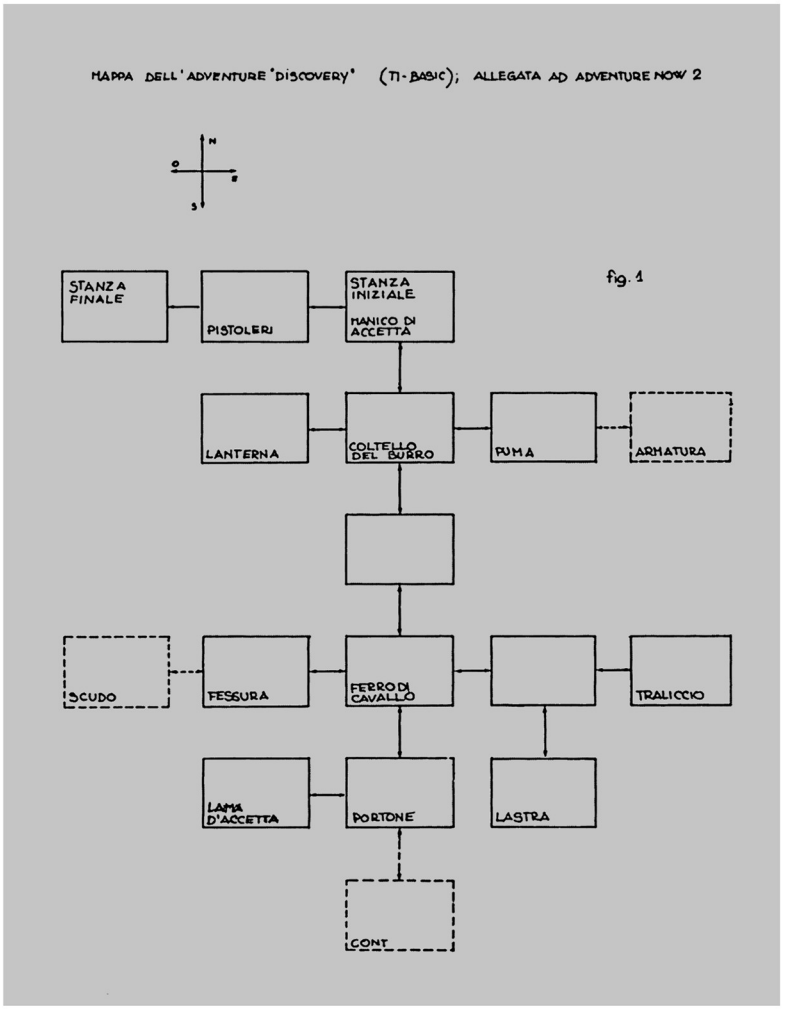
1 - Prendere "manico accetta": andare a SUD 3 volte;
2 - Prendere il "ferro di cavallo": andare a SUD poi a OVEST;
3 - Prendere la "lama d'accetta" e dirigersi al "traliccio";
4 - Toccare il "ferro";
5 - Unire "manico" (dell'accetta);
6 - Tagliare il "traliccio";
7 - Prendere la "dinamite" caduta;
8 - Dirigersi al "portone";
9 - Abbattere il "portone";
10 - Guardare: si è aperta una porta a SUD;
11 - Andare a SUD, lasciare manico e lama, prendere il "cent";
12 - Andare dove c'è la fessura e infilarci il "cent";
13 - Si apre una porta ad OVEST, andate e prendete lo "scudo"
14 - Adesso andate a prendere la "lastra";
15 - Tornate dove c'era il coltello da burro e affilatelo (avete la lastra!);
16 Entrate nella baracca con il puma e uccidetelo;
17 - Potete andare a OVEST adesso;
18 - Prendete l'armatura. Ora con scudo e armatura potete eludere i killers e completare l'avventura.

Ed ora, dopo la soluzione, continuiamo il nostro viaggio nel mondo degli ADVENTURE-GAME.
La scorsa volta, se vi ricordate, avevamo spiegato in maniera piuttosto generale cosa è un adventure - game e come si gioca.
Questo mese invece esamineremo la struttura che dovrà possedere il programma. Per comodità divideremo, o meglio sezioneremo, il nostro programma in tanti sottoprogrammi più semplici, ognuno con un ben determinato compito. Faremo quindi quello che i programmatori chiamano "sviluppo TOP-DOWN".
Distingueremo i seguenti blocchi:

a) MAIN PROGRAM
o programma principale, il quale ha il compito di gestire i vari sottoprogrammi.

b) SUB INPUT
si occupa della gestione dell'input, cioè del comando impartito dall'utente.

c) SUB LOGICA
controlla la logica dell'azione,



ed in caso si verifichi una condizione non significativa, la segnala.

d) SUB MESSAGGI
fornisce messaggi esplicativi inerenti all'azione corrente.

e) SUB DATA
contiene l'intero "vocabolario", le descrizioni delle stanze e quelle degli oggetti.
In figura 2 viene mostrato uno schema di flusso che vi chiarirà la funzione di ogni blocco nel programma.

SUB FINE
Questo blocco è molto probabilmente il più semplice da implementare. Generalmente, il suo contenuto si riduce ad un semplice (e molto irritante) messaggio di cordoglio per la prematura fine, oltre alle istruzioni necessarie per ricominciare l'avventura da capo.
Il sottoprogramma FINE è richiamato da quello LOGICA.

SUB MESSAGGI
Anche questo blocco, come quello precedente, non presenta molte difficoltà per la propria stesura. In esso sono contenuti tutti i messaggi inerenti all'azione corrente. Si tratta per lo più di segnalazioni di condizioni di errore nello svolgimento dell'azione, o di veri e propri indovinelli, il cui scopo è quello di far lavorare la "materia grigia" dell'utente.
Appare chiaro, allora, che sarà principalmente il sottoprogramma LOGICA a richiamare il SUB MESSAGGI, seguito da quello SUB INPUT.

SUB INPUT
Il compito di questo sottoprogramma è quello di ricevere, e in un certo senso interpretare, il comando impartito dal giocatore. La maggior parte dei programmi in circolazione accetta come input 2 parole, generalmente VERBO + NOME, ciò significa che bisognerà dividere l'input in due parti: una contenente il verbo, l'altra l'oggetto. Anche noi, per comodità, useremo questo sistema, conveniente sotto molti punti di vista. Così facendo, infatti, sarà molto più facile ricercare, successivamente, le parole memorizzate (verbo e nome) nelle tabelle dei verbi e dei nomi.
Inoltre, se useremo tale convenzione, sarà possibile accettare anche parole abbreviate, velocizzando notevolmente il tutto. Potremo poi fare in modo di impartire i comandi più comuni (guarda, vai a nord, sud, est, ovest, inventario) utilizzando una sola lettera (di solito quella iniziale) semplificando ulteriormente la gestione dell'input.
A questo punto non resta altro che enunciare l'algoritmo e

T99IUC.it

provare a scrivere la parte BASIC che lo traduce.
L'algoritmo è il seguente:
Passare il controllo all'utente che potrà impartire il comando; battuto l'ENTER, il calcolatore lo riceverà e lo memorizzerà nella variabile MEMO$. Se la lunghezza del comando sarà uguale a uno (LEN (MEMO$) = 1) lo cercherà in una apposita tabella, altrimenti effettuerà la separazione dell'input in due parole distinte (verbo + oggetto) e controllerà in apposite tabelle.
N.B. Per separare sarà necessario almeno uno spazio tra una parola e l'altra''.

Come avrete notato il tutto sembra molto complicato (ma non spaventatevi!!!); sarà sufficiente seguire passo per passo quanto enunciato dall'algoritmo per scrivere senza troppi problemi la nostra routine di INPUT.
In figura 3 troverete il diagramma di flusso che traduce l'algoritmo in questione: probabilmente chiarirà ulteriormente quanto affermato. Ed ora passiamo alla routine BASIC vera e propria:

```
100 INPUT
''Abbreviate a quante lettere?'': LUN 110 INPUT
''Comando?'':
MEMO$
120 IF LEN (MEMO$) = 1 THEN 200
130 VERBO$ = SEG$ (MEMO$,1,LUN)
140 FOR I = LUN + 1 TO LEN (MEMO$)
150 K$ = SEG$ (MEMO$, I, 1)
160 IF K$ = '' '' THEN NOME$ = SEG$ (MEMO$, I + 1, LUN)
170 NEXT I
180 PRINT VERBO$,NOME$
190 STOP
200 REM
(controllo in tabella)
210 STOP
```

■

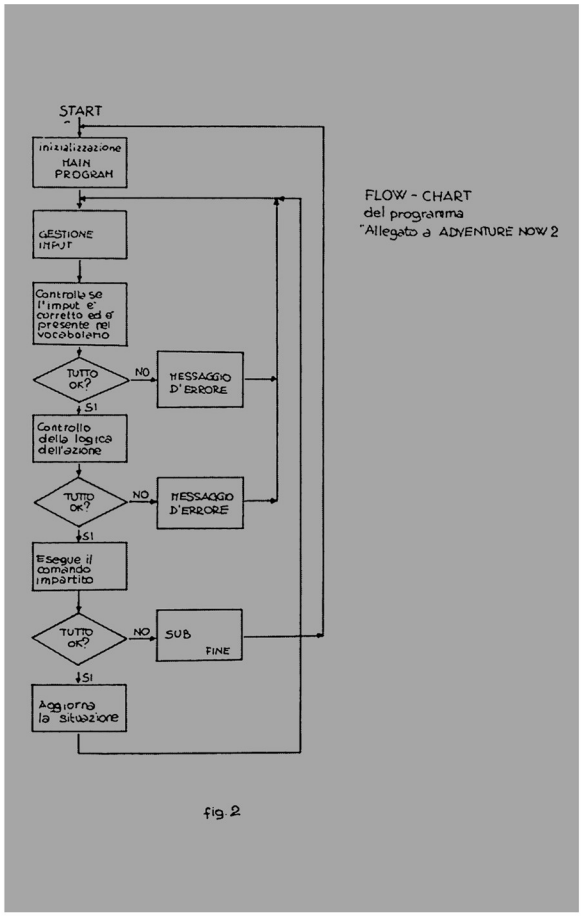


fig. 2

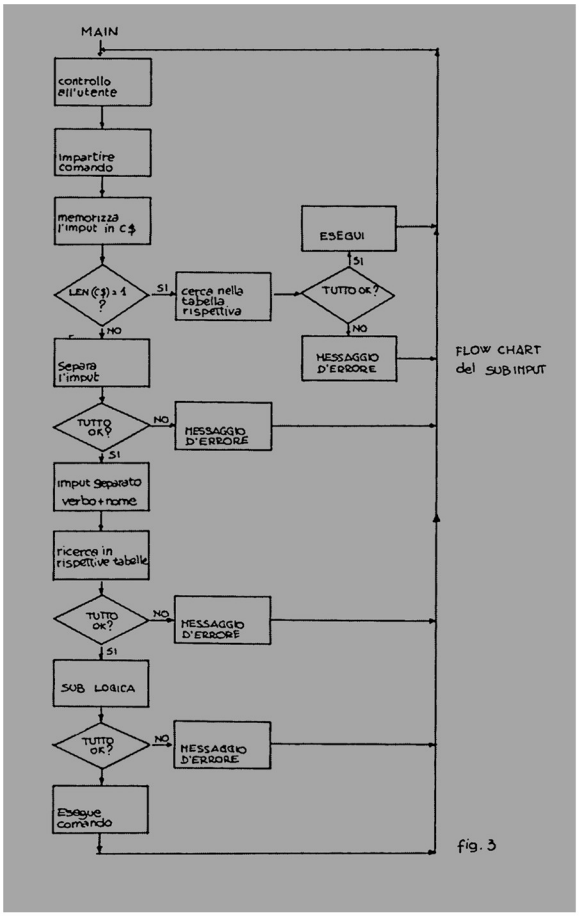


fig. 3

LE MEMORIE DI MASSA

di MARCO SQUINTANI

Chi ci segue dalla prima puntata avrà avuto modo di approfondire meglio l'argomento, o, magari, di scoprire che il computer che utilizza è dotato di una memoria interna volatile (la RAM) che permette di conservare istruzionl e dati solo finché il "personal" è acceso. Per conservare invece grosse quantità di dati è necessario ricorrere a memorie permanenti esterne, dette, appunto, memorie di massa.
Per il TI99 esistono due sistemi. Il primo, che è anche il più diffuso ed economico, è l'unità a cassette magnetiche; l'altro è l'unità a dischi.

L'unità a cassette magnetiche

Il nastro magnetico, che fino a pochi anni fa non aveva rivali nel campo delle memorie di massa, resta — come abbiamo già detto — il sistema che offre il miglior rapporto prestazioni/prezzo, sebbene con alcune limitazioni. Questa unità, che ognuno di voi avrà sicuramente sotto mano, è composta da un semplice registratore per musicassette, di una qualsiasi marca e dimensione, collegato tramite il cavetto-interfaccia. L'informazione viene immagazzinata sul nastro magnetico della cassetta sotto forma di suoni (che potrete sentire distintamente aumentando il volume del monitor o televisore). Un segnale audio con frequenza determinata rappresenta le cifre binarie 0 e 1 e la funzione dell'interfaccia è proprio quella di codificare e decodificare dati e segnali.
I dati vengono letti e scritti alla velocità di 300 Baud (1 Baud = 1 carattere o bit al secondo) in maniera sequenziale: è proprio questo il principale svantaggio del sistema a cassette magnetiche. Per leggere un programma registrato verso la fine della cassetta è infatti necessario attendere lo scorrimento del nastro fino al punto desiderato, come la ricerca di un qualsiasi brano musicale. Si può tuttavia facilitare tale operazione avendo cura di segnare sulla cassetta o su un foglietto di carta il conteggio del nastro stesso e arrivare poi al punto voluto con i tasti di ricerca veloce (F.F. e REW). A questo inconveniente, che può costituire un grosso problema per chi abbia la necessità di caricare frequentemente lunghi programmi o file di dati, si contrappone, comunque, il basso costo del dispositivo e la possibilità di utilizzare un comune registratore a cassetta di tipo domestico, purché abbia i tre canali di collegamento di lettura, scrittura e controllo a distanza.
Per evitare perdite di dati a causa delle brusche fluttuazioni di volume del segnale audio proveniente dal computer, usando un registratore dotato di controllo automatico del livello di registrazione, il nostro TI 99 emette i dati da memorizzare in blocchi separati, preceduti da una nota continua della durata di qualche secondo. Il piccolo stratagemma permette così al circuito di controllo automatico di stabilizzare il livello di entrata, migliorando l'uscita dei dati stessi una volta attivato il tasto di ascolto (PLAY).
Nonostante ciò, molti lettori si lamentano di non riuscire a caricare correttamente i programmi contenuti nelle nostre ed altre cassette, attribuendo colpe inesistenti ora al computer, ora agli esecutori materiali della registrazione (noi, tanto per intenderci). Compito che ci prefiggiamo nel prossimo numero sarà allora quello di fornirvi una completa informazione sull'uso corretto del registratore, sia in ascolto che in registrazione. A presto. ■

ALTA RISOLUZIONE PER MINI MEMORY

di RICCARDO ROSSI

Il microprocessore grafico del TI-99/A4 è universalmente riconosciuto come il migliore del suo genere. Chi ha la fortuna di possedere il logo o il pascal si sarà reso conto di come questa macchina possa gestire la grafica ad alta risoluzione. Purtroppo per chi non ha uno di questi due linguaggi l'accesso all'alta risoluzione è vincolato all'uso dell'assembler, (sempre vincolante la presenza di una scheda RAM).
In assembler sorge però un piccolo problema; se il nostro scopo è quello di tracciare il grafico di una funzione potremmo avere delle difficoltà nel momento in cui dobbiamo inserire la funzione da calcolare. Per fare i conti non c'è dubbio che il BASIC sia più adeguato dell'assembler. Perché allora non scrivere delle routine assembler che ci permettano di usare l'alta risoluzione in BASIC?
A questo punto, probabilmente, tutti i possessori della mini memory avranno già caricato il programma con il comando L dell'easy bug, e saranno smaniosi di poterlo usare.
Un attimo di calma; prima bisogna fare un paio di precisazioni sui limiti del programma:

1) l'alta risoluzione richiede 14 k di VDP RAM; questo limita la memoria disponibile in basic a soli 2 k.

2) il basic non riconosce l'alta risoluzione quindi bisogna forzare dei puntatori BASIC con il co mando:

```
NL CALL LOAD
(-32658, 56, O"' ,
-32732, 55, 248)
```

3) Alcune funzioni matematiche utilizzano una zona di memoria VDP come area di lavoro, questo comporta un disturbo del disegno. Per aggirare questo inconveniente sono state create due routine:

```
CALL LINK ("FUNC1")
e
CALL LINK ("FUNC2")
```

che vanno utilizzate rispettivamente prima e dopo di effettuare dei calcoli. Lo scopo delle due routine è di salvare la parte di disegno che può essere distrutta per poi poterla rimettere a posto.

4) Al termine di un programma il basic non riporta automaticamente la grafica a posto, per cui prima di terminare un programma è bene fare eseguire questa istruzione:

```
NL CALL LINK ("VWTR", 0,0)
```

Vediamo ora quali sono e cosa fanno questi sottoprogrammi per la gestione della grafica ad alta risoluzione (i parametri racchiusi fra parentesi quadra ([]) sono opzionali).
Al programma assembler segue un piccolo programma BASIC che dà un'idea di quello che è possibile realizzare con queste routine.

CALL LINK ("HGR") entra in alta risoluzione
CALL LINK ("CLEAR") cancella la pagina grafica
CALL LINK ("PLOT", X, Y, CP [CS] accende il punto di coordinate X,Y di colore CP e con sfondo CS
CALL LINK ("LINE", X1, Y1, X2, Y2, CP [CS] traccia una retta tra i punti di coordinate X1, Y1, e X2, Y2, di colore CP e con sfondo CS
CALL LINK ("TEST", X,Y,E,F,B) controlla il punto di coordinate X,Y e restituisce
in T, 1 se il punto è acceso, 0 se è spento.
in F il codice del colore del punto
in B il codice del colore dello sfondo
CALL LINK ("TEXT" A$, [RIGA, COLO, CP, CS]) scrive la stringa A$ nella posizione RIGA, COLO, di colore CP e con sfondo CS
CALL LINK ("SAVE") salva su registratore a cassette il disegno
CALL LINK ("LOAD") carica un disegno precedentemente salvato su nastro
CALL LINK ("VWTR", REG, VAL) scrive nel VDP registro N° REG il valore VAL
CALL LINK 5 ("FUNC1") da eseguire prima del calcolo di funzioni
CALL LINK ("FUNC2") da eseguire dopo il calcolo di funzioni
CALL LINK ("OR") dopo questo comando quando si tenta di accendere un punto già acceso questo non varia il suo stato
CALL LINK ("XOR") dopo questo comando quando si tenta di accendere un punto già acceso questo si spegne, mentre se il punto è spento viene acceso
CALL LINK ("POINT", X,y) sullo schermo compare un cursore che può essere spostato con i Joystick o i tasti freccia. Premendo il pulsante di sparo o enter vengono restituite in x e y le coordinate del centro del cursore.

È uscito il primo numero di

SOFTWARE & SOFTWARE

Downloaded from www.ti99iuc.it

una nuova rivista dedicata al TI-99 4/A, dove troverete i listati di alcuni tra i migliori programmi per il vostro computer

RICHIEDETELA IN EDICOLA COSTA SOLO Lire 3.000

SUPERCALC

di BOTTIN GIANPAOLO

Questo programma è stato convertito da uno analogo per lo ZX SPECTRUM ed è in grado di calcolare:
1) radici quadrate di numeri positivi minori di 1E + 16 fornendo un risultato con un massimo di 110 cifre decimali
2) M.C.D.; calcola il massimo comune divisore di due o più numeri (fino a sei per ragioni di spazio sullo schermo)
3) m.c.m.; calcola il minimo comune multiplo tra due o più numeri
4) semplificazione di frazioni; divide sia il numeratore che il denominatore di una frazione per i loro divisori comuni
5) ricerca la frazione generatrice di un numero decimale anche periodico, viene richiesta prima la parte intera poi l'antiperiodo (cioè la parte tra la virgola e il periodo), se non esiste basta semplicemente premere ENTER, quindi viene richiesto il periodo (la parte che si ripete all'infinito ed è segnata con un trattino) che può anche non esistere, in tal caso è sufficiente premere ENTER'
6) divide due numeri tra loro dando il risultato con il numero di cifre decimali desiderato (non più di 150 poiché le altre non ci starebbero sullo schermo)
7) scompone un numero in fattori primi, se il numero è piuttosto grande impiega un certo tempo a trovare tutti i fattori.
Durante la traduzione di questo programma abbiamo dovuto affrontare notevoli problemi causati dalla mancanza di istruzioni fondamentali che sono per fortuna presenti nel più completo EXTENDED BASIC, ad esempio abbiamo dovuto scrivere delle routine di ACCEPT AT e di DISPLAY AT per evitare lo scrolling dello schermo durante le operazioni di input e di output.
Un ulteriore problema nel convertire programmi da un altro computer in TI BASIC è dato dalla mancanza di istruzioni grafiche quali PLOT, DRAW, LINE, queste ultime sono particolarmente difficili da tradurre, però, sapendo qual è il risultato finale di tali istruzioni (ad es. la costruzione di un quadrato) si possono definire dei caratteri grafici da stampare poi sullo schermo in posizioni opportune.
Ad esempio nella routine di calcolo della radice quadrata vi erano delle DRAW per disegnare il simbolo della radice, in questo caso è stato sufficiente definire i caratteri 123, 124, 125 e stamparli con delle HCHAR.
Un'altra difficoltà riguardante la grafica è costituita dai caratteri grafici definiti dall'utente: per lo SPECTRUM occorrono 8 numeri decimali tra 0 e 255 per definire un carattere di 8 x 8 pixel, al contrario sul nostro TI occorre una stringa esadecimale di 16 caratteri; per la conversione è sufficiente trasformare i numeri da decimali in esadecimali quindi inserirli in una stringa uno di seguito all'altro, stringa che può essere usata in una CALL CHAR.
Ad esempio

```
 10 FOR B=I TO 8
 20 READ X
 30 POKE USR "A" + B,X
 40 NEXT B
200 DATA 224, 128, 224,
      166, 230, 0,0,0
```

definisce il carattere grafico A, i data si possono convertire così (utilizzando un qualunque metodo di conversione di decimale a esadecimale):

224 — — — E0
128 — — — 80
224 — — — E0
166 — — — A6
230 — — — E6
0 — — — 00
0 — — — 00
0 — — — 00

e basta fare CALL CHAR (129, "E080E0A6E6") per definire in carattere 129 corrispondente a CTRL A.

Uno dei principali problemi nel tradurre un programma in TI BASIC è la mancanza di linee multistatement che andranno riscritte su più righe, ad esempio:

```
10 IF A=B THEN PRINT A:
   PRINT B: C=A+B
```

occorre riscriverla invertendo la condizione della IF:

```
10 IF A <> B THEN 20
11 PRINT A
12 PRINT B
13 C=A+B
20 ...
```

Nei casi più semplici si può procedere come abbiamo visto ma esistono casi in cui solo con intricate circonvoluzioni è possibile mantenere inalterata la logica del programma.

Altri problemi nascono dalle operazioni sulle stringhe, l'operatore + deve essere sostituito con &, inoltre:

A$ (A TO B) — — — SEG$ (A$, A, B—A+1)
A$ (TO B) — — — SEG$ (A$, 1, B)
A$ (A TO) — — — SEG$ (A$, A, LEN (A$)—A+1)

Passiamo ora ad esaminare il programma senza comunque scendere nei dettagli di ogni routine matematica. ■

100 - 340	inizializza le variabili e definisce i caratteri grafici
350 - 480	stampa il menù principale
490 - 580	rimane in attesa che venga premuto un tasto e nel frattempo fa lampeggiare il cursore
590 - 610	se è stato premuto un tasto da 1 a 7 salta alla relativa routine
620 - 630	cancella le ultime 6 righe dello schermo e torna in attesa di un input
640	inizio della routine per la radice quadrata
650 - 820	viene richiesto prima il radicando quindi il numero di cifre decimali desiderato
830 - 970	stampa il simbolo della radice quindi salta alla subroutine che effettua il calcolo vero e proprio, al ritorno attende che venga premuto un tasto
980	routine per il calcolo dell'M.C.D.
990 - 1210	richiede i numeri tra i quali deve cercare l'M.C.D.
1220 - 1270	calcola l'M.C.D. utilizzando la subroutine 4030 che effettua il calcolo vero e proprio
1280 - 1350	stampa l'M.C.D. e attende che sia premuto un tasto
1360	inizia la routine che calcola l'm.c.m.
1370 - 1590	richiede i numeri per i quali deve cercare l'm.c.m.
1660 - 1730	stampa l'm.c.m. e attende che venga premuto un tasto
1740	semplificazione di frazioni
1750 - 2090	viene richiesto il numeratore e poi il denominatore quindi vengono stampati insieme alla linea di frazione
2100 - 2320	richiama la subroutine 4170 che effettua il calcolo quindi stampa il risultato, se è intero non viene stampata la linea di frazione, quindi attende che venga premuto un tasto
2330	routine di ricerca della frazione generatrice
2340 - 2730	vengono richiesti i dati in input, parte intera, antiperiodo, periodo quindi vengono stampati
2740 - 2940	effettua il calcolo mediante la subroutine 4240 quindi stampa i risultati e attende che venga premuto un tasto
2950	calcola la divisione tra due numeri
2960 - 3310	richiede numeratore e denominatore che vengono stampati, inoltre viene richiesto il numero di decimali desiderato
3320 - 3360	salta alla subroutine 4310 che effettua il calcolo quindi attende che venga premuto un tasto
3370	routine per la scomposizione in fattori
3380 - 3530	viene richiesto il numero da scomporre
3540 - 3580	salta alla routine di calcolo e attende che venga premuto un tasto
3590 - 4020	subroutine che effettua il calcolo della radice quadrata
4030 - 4100	subroutine che calcola l'M.C.D.
4110 - 4160	subroutine che calcola l'm.c.m.
4170 - 4230	subroutine per la semplificazione di frazioni
4240 - 4300	subroutine che effettua la divisione tra due numeri
4530 - 4900	subroutine che scompone un numero in fattori primi
4910 - 5010	subroutine di PRINT AT, viene stampato il contenuto della variabile A$ alle coordinate Y1 e X1
5020 - 5380	subroutine di ACCEPT AT, aspetta l'input alle coordinate Y1 e X1 e lo mette nella variabile A$
5390 - 5400	data per la stampa del menù

TI WATER SKIING

di DANIELE CATALFAMO

ANALISI DEL LISTATO

100 - 320	Definizione e stampa della schermata iniziale
330 - 350	Inizializzazione primo ciclo
360 - 380	Routine primo schermo cioè quello degli scogli
390 - 430	Secondo schermo
440 - 500	Terzo schermo
510 - 550	Quarto schermo
560 - 570	Quinto schermo
580	Sub creazione sprite per il secondo schermo
590 - 600	Sub passaggio allo schermo successivo
610 - 850	Sub inizializzazione colori e caratteri per il prossimo schermo
860 - 870	Sub movimento sciatore
880 - 890	Sub lettura tastiera
900 - 910	Sub movimento motoscati
920 - 950	Sub movimento squali
960	Sub collisione
970 - 990	Sub game over
1000 - 1020	Sub messaggio di errore
1030 - 1040	Sub ripetizione in caso si siano finiti tutti gli schermi

Per utilizzare i Joysticks modificare la seguente linea:

880 ***SUB KEYST (M):: CALL JOYST (1,K,S):: A***=S*3/2: : CALL POSITION (÷1,Y,X): : B=4* (X>15) +8* (K=4)*(X<220): : CALL MOTION (÷1,A,B, ÷2,A,B)

Questa volta siamo alle prese con un gioco acquatico. Dobbiamo infatti guidare un coraggioso sciatore, trainato da un potente motoscafo. La missione consiste nell'affrontare e superare i vari ostacoli che si presenteranno ad ogni schermo; e sono ben cinque e tutti diversi!!

Primo schermo: è il più facile, dovremo fare solo attenzione a schivare dei tronchi e dei piccoli icebergs che procedono in linea retta.

Secondo schermo: iniziano i guai. Numerose barche a vela ondeggiano sul video, ma schivarle non sarà impresa facile.

Terzo schermo: le cose si complicano. Compaiono piccoli motoscafi aventi un unico scopo: speronarvi.

Quarto schermo: se siete arrivati fin qui, allora coraggio. Infatti dovrete affrontare cinque feroci squali, pronti a sbranarvi.

Quinto schermo: l'ultima fatica. Numerose mine galleggianti cercheranno di ostacolarvi per impedirvi di completare il ciclo del gioco.
Ed ora prima di finire, due appunti. Il primo è per la grafica, molto ben riuscita. Inoltre gli ultimi due gruppi di caratteri contengono un set di simboli grafici che vi sarà utilissimo per costruire scenari e sfondi. Il secondo appunto è per la struttura del programma.
Anche in questo caso, ci siamo sforzati di strutturare il programma. Ricordate: Strutturazione = Buona Programmazione. Infatti strutturando il programma, questo viene suddiviso in tanti piccoli problemi, che faciliteranno la comprensione del programma principale, e che potrebbero essere riciclati in futuro. ■

BIBLIOTECA

di PAOLO CIVARDI

Vi proponiamo stavolta un programmino semplice ma di estrema utilità.
Questo programma vi permetterà di tenere in ordine i libri che costituiscono la vostra biblioteca, di organizzare le vostre ricerche e, persino, di tenere d'occhio quella stirpe calamitosa di amici che richiedono in prestito dei libri fidando nell'usanza dell'usucapione; dopo vent'anni d'uso ritengono i volumi prestati definitivamente di loro proprietà.
Questo programma, strutturato com'è sulla lettura di sole tre variabili, ben si presta ad essere facilmente modificato secondo i vostri bisogni: potete, per esempio, assegnare ad una variabile, temporaneamente, la data del prestito e il nome del reo colpevole del ritardo nella restituzione, o quanto altro le vostre necessità vi possono consigliare.
Nella loro forma attuale le tre variabili sono impiegate per la lettura del nome del volume, per identificare l'autore e per l'esatta collocazione del libro sullo scaffale della vostra biblioteca.
Tutto ciò tramite la lettura di DATA messi in fondo al programma.
Suggeriamo, se possibile, di cercare di collocare i numeri degli scaffali in ordine crescente per velocizzare al massimo la ricerca.
Valgono poi le solite scritte sul manuale del vostro calcolatore per l'impiego delle istruzioni DATA, rispettare la separazione dei dati con virgole ecc. e ricordate poi di porre per ultimo sempre la parola FINE, che viene impiegata dal programma per riconoscere il termine dell'operazione di lettura.
Il numero dei dati disponibili è, per comodità, fissato preliminarmente a 1000; numero ritenuto sufficiente per la gestione di una normale biblioteca familiare ma nel caso possediate qualche cosa simile, dimensionalmente, all'Archivio di Stato, aumentate pure, fin quanto è possibile la variabile L e poi telefonateci, volentieri ci faremo prestare da voi alcuni libri nel tentativo di aiutarvi a contenere il problema. Ultimo avviso per il corretto uso del programma: in fase di visualizzazione dei testi noterete che ogni tanto il programma arresterà lo scroll del video; cio è stato fatto volutamente per darvi il tempo di leggere comodamente quanto scritto. Basterà pertanto battere la barra spaziatrice per proseguire, quando lo riterrete opportuno, la lettura. ■

TI-PUFFI

di DANIELE CATALFAMO

Questo programma fa parte di una prima triade di ADVENTURES che saranno basati sugli algoritmi spiegati dagli articoli della rubrica PROGRAMMAZIONE.
Sarete trascinati nel "simpatico" mondo di quei nanetti blu che hanno divertito grandi e piccini: I PUFFI!!!
Dovrete quindi aiutare i nostri piccoli amici a ritrovare la loro unica compagna: LA PUFFETTA.
La ricerca inizierà proprio dal loro villaggio, ed è chiaro che i sospetti cadono tutti sul perfido mago GARGAMELLA.
Sarà quindi opportuno arrivare al suo castello per cercare ulteriori indizi. Avete a disposizione tre strade:
1) la prima attraversa il DESERTO;
2) la seconda si snoda sui MONTI;
3) la terza lungo il MARE.
Ogni strada ha la propria difficoltà e pericoli da superare. Inoltre, il CASTELLO, sorge su di un'isola. Per cui dovrete sforzare la vostra materia grigia per giungervi. Ulteriori informazioni sono contenute all'interno del programma insieme all'intero VOCABOLARIO. Come la volta scorsa, inoltre, la soluzione la troverete sul prossimo numero di TI 99 NEW SOFT.

Funzionamento

Data la complessità del Programma, si è preferito come linguaggio il TI-EXTENDED BASIC, il quale offre la possibilità di righe multi-statement e una serie di istruzioni non presenti nel semplice TI-BASIC. Inoltre data l'eccessiva occupazione di memoria il programma è stato spezzato in due.
— La prima parte contiene la presentazione e le istruzioni necessarie per giocare oltre al prologo della storia.
— La seconda parte contiene l'avventura vera e propria.
I due programmi vanno caricati uno per volta; il primo dopo la presentazione e le istruzioni lancerà automaticamente l'altro che dovrà essere allora caricato. Non resta quindi che augurarvi BUON DIVERTIMENTO e darci appuntamento alla prossima avventura con "ADVENTURE NOW".

P.S. Invito nuovamente tutti i possessori della cartuccia RETURN TO PIRATE ISLE a comunicarci il punto fino a cui sono giunti (e come!!!), in quanto noi siamo arrivati ad un punto morto. Di questa cartuccia comunque parleremo più ampiamente un'altra volta.
Voglio però annunciarvi che molto presto parleremo degli ADVENTURE della TEXAS, che stiamo per ora traducendo, e delle loro soluzioni.
Invitiamo, quindi, di nuovo tutti coloro che ne avessero risolto uno o fossero giunti a buon punto di scriverci.
Ringraziamo tutti quanti anticipatamente per la collaborazione che ci verrà fornita. ■

ALGOL

di DANIELE CATALFAMO

Il gioco che l'M.M.G. vi presenta questo mese è la versione per TI 99 + Extended Basic del noto gioco ALGOL. Questo programma simula un combattimento spaziale di notevole complessità. Durante il gioco (diverso ogni volta) si verificano differenti situazioni.

IL GIOCO:

Dovete manovrare l'astronave Enterprise per eliminare i vascelli Klingon, vostri nemici, da un settore di 8x8 caselle al cui interno si trovano anche 10 zone meteoritiche.
Disponete di una scorta di 100.000 unità di energia. Ogni volta che vorrete aprire il fuoco basterà specificare le coordinate di tiro e scegliere il livello di energia. Tenete presente che con 5.000 unità di energia potete distruggere le navi ammiraglie, mentre per i ricognitori ne bastano solo 1.000.
Le informazioni sullo schermo sono numerose e sufficientemente chiare. Un dato molto importante che dovete tenere sempre d'occhio è l'*energia residua:* se scende sotto il livello minimo il gioco viene interrotto e per voi è tutto da rifare.
Sullo schermo sono inoltre presenti:

— Il settore di 8x8, simile a quello della battaglia navale, con indicate le coordinate su cui vengono segnati i tiri effettuati.
— Il Radar, che mostra ricognitori e zone meteoritiche nella zona limitrofa al punto di fuoco.
— L'esito del vostro tiro.
— Time, che è un contasecondi: avete circa 10 secondi per effettuare il nuovo tiro, altrimenti la situazione verrà aggiornata automaticamente con la perdita di 500 unità di energia.

Ricordate bene però che i ricognitori si muovono per cui quando voi sparate alle coordinate indicate dal Radar non è detto che il ricognitore sia ancora lì a farsi colpire! Comunque per vostra fortuna essi non invertono mai la rotta indicata.

Inoltre a complicare le cose ci si mettono anche le ammiraglie. Queste sono protette da un campo di forze che le rende invisibili al radar, rappresentano quindi un notevole rischio per il capitano imprudente: colpirle con 1.000 unità di energia significa non solo non provocare alcun danno, ma anche segnalare la posizione dell'Enterprise che si troverà ben presto a mal partito.
Avete a disposizione due errori con conseguenti danni e rispettive perdite di 6.000 e 12.000 unità di energia, dopodiché al 3° errore verrete distrutti. Comunque, ogni volta che distruggete un'ammiraglia con 4 tiri consecutivi ricevete un bonus di ben 20.000 unità di energia.
Tenete inoltre ben a mente le zone meteoriche in quanto nè le navi ammiraglie nè i ricognitori si possono sovrapporre ad esse, così facendo risparmierete preziosa energia (evitando tiri inutili). Ogni volta che vorrete inserire nuove coordinate di tiro basterà premere un tasto, prima che il Time arrivi a 10, poi inserire la lettera e poi il numero che indicano la casella da colpire; inserite quindi l'energia battendo 1 o 5 per 1.000 o 5.000 e aspettate il responso.
Noi la nostra parte l'abbiamo fatta, ora tocca a voi, novelli «Kirk», impegnarvi nella lotta contro i Klingon per una disperata battaglia. Siate comunque sempre prudenti e attenti a non farvi degradare!!!

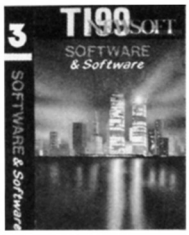
■

TI/99 NEWSOFT la prima rivista con cassetta

Se avete perso qualche numero, o se non la conoscete ancora approfittate delle eccezionali condizioni che TI-99 NEWSOFT vi offre in occasione del suo primo compleanno.

GIUGNO '84
Basic-Extended: i trucchi, i misteri; Assembler e L/M.
Giochi: Othello, Nibbler, Cubo Magico, Difesa, Galaxy II, Incontri Ravvicinati.

SETTEMBRE '84
Assembler, Mini Memory: come utilizzare questo modulo che apre le porte sul mondo dell'Assembler.
Inizia la rubrica ''Impariamo a costruire un arcade game: gli sfondi mobili''.
Giochi: Tennis, Crazy Koala, Pac-Man, Slalom Gigante.



OTTOBRE '84
Char Def. Programma in Extended Basic con il quale è possibile definire i caratteri grafici; istruzioni di I/0; Extended: sprites. Statistica. Bioritmi. Equo Canone.
Giochi: Bomb on New York, Assedio, Yatzee.

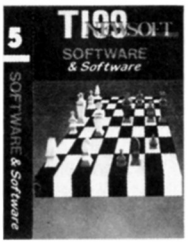
NOVEMBRE '84
Modulo Mini Memory (mappa RAM). Come simulare gli operatori logici in Basic. Sfondi Mobili. Statistica.
Giochi: Guardia e Ladri, Lander, Space Invaders, Coyote, Gran Prix.

DICEMBRE '84
Basic: gestione dei files su cassetta; Didattica. Merge.
Giochi: Star Wars, Torre di Hanoi, Notte di Natale, Doni Celesti

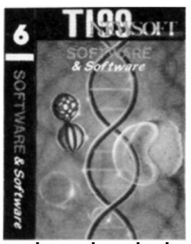
Per le richieste utilizzare l'apposita cartolina

per il tuo TI-99/4A COMPIE UN ANNO!!!

1 Numero arretrato	L. 15.000
4 Numeri arretrati	L. 40.000
1 Copia saggio a nostra scelta	L. 5.000
Tutti i 10 numeri già usciti	L. 80.000



GENNAIO '85
Hardware: come usare la memoria da 32K esterna.
Assembler e L/M: il Multicolor. Mini Memory: ROM.
Archivio scacchistico. Statistica.
Giochi: Battaglia Navale, Poker, The Cave.



downloaded

FEBBRAIO '85
Extended + 32K: il modo per programmare in L/M.
Merge: Merge per la Mini Memory. Statistica. Didattica.
Giochi: Il castello notte scura, Lunar Lander, Rockfeller.
Programma musicale.

from

MARZO '85
Come si può elaborare un programma Assembly in L/M.
Writer Loader: carica in memoria
sottoprogrammi in L/M. Statistica. MCP: Sistema
di archiviazione programmi. Giochi: Hot Dog, Strike Defence,
Missile Command, Nibbler in Basic.

www.ti99iuc.it

APRILE '85
Extended Basic + Espansione 32K. Le ROM, PROM
e EPROM, cosa sono e come lavorano. Arcade Game:
i comandi. Mini Memory; Alta Risoluzione. Didattica.
Recensioni: Up Periscope.
Giochi: Robotime, Avventura nel calcolatore.

MAGGIO '85
Basic: come sfruttare i tasti di controllo della tastiera.
Extended + 32K: come ottenere il text mode + tre nuove
istruzioni. TI-99-4/A: esaminiamo la ''carrozzeria''
del TI-99. Statistica. Aiuto alle decisioni: un programma
che vi guiderà all'acquisto di un'autovettura.
Recensioni: Snow Trek. Giochi: 3-D Monsters Manor.
Bouncer, Album: per bambini in età prescolare.

inserita nella rivista.

LIFE

di ERMINIO TRILLO

Il programma "Life" non è altro come s'intuisce dal nome che una versione per il TI-99/4A del celeberrimo algoritmo inventato dal matematico John Horton Conway sul finire degli anni sessanta, basato su alcuni studi teorici di uno dei padri dell'informatica, John von Neumann.
Com'è noto, il programma, al quale lo stesso CONWAY ha dato il nome di "Life" (Vita), è il prototipo dei giochi di simulazione, è perciò un gioco in senso piuttosto lato. Si tratta, infatti, della rappresentazione di alcuni punti che appaiono o scompaiono sullo schermo o sul monitor in base a determinate regole. Le immagini delle varie configurazioni che si succedono l'una all'altra simulano lo sviluppo di un sistema di immaginari organismi unicellulari, rappresentandone le relative generazioni.
Naturalmente il mondo abitato da questi organismi è un mondo piatto, che conosce due sole dimensioni nel tempo e nello spazio: lo spazio consiste di un numero finito di punti possibili, ciascuno dei quali è individuato dalle coordinate COLONNA (A-P) e RIGA (1-16); il tempo non scorre in modo continuo, ma discreto, passando direttamente da una generazione di organismi alla successiva, senza "istanti" intermedi. Il trascorrere del tempo è dimostrato, quindi, soltanto dai mutamenti di stato degli organismi, che possono essere "vivi" o "morti" e sono raffigurati, nel primo caso, con una casella occupata, — che corrisponde ad un punto del nostro spazio — nel secondo caso con una casella vuota.
Le regole che determinano i cambiamenti di stato degli organismi, che possono essere facilmente cambiate modificando le linee 990-1180 del programma, sono le seguenti:
1) se nelle otto caselle adiacenti alla casella considerata vi sono due organismi vivi, l'eventuale organismo presente nella casella sopravviverà nella generazione successiva, ma non ne nascerà un altro se la casella era vuota;
2) se vi sono tre organismi vivi ne nasce uno nella casella considerata, o vi sopravvive quello eventualmente presente;
3) se vi sono meno di due o più di tre organismi vivi, l'eventuale organismo presente nella casella considerata muore, rispettivamente, per isolamento o per sovraffollamento.
Poiché gli stati in cui si possono trovare gli organismi sono soltanto due, è facile inserire ogni generazione in una matrice, i cui elementi avranno valore 0 per "morto" ed 1 per "vivo". Si avrà poi cura di riprodurre in una nuova matrice il contenuto della generazione appena esaminata prima di passare alla successiva, ad evitare che gli stati degli organismi della nuova generazione influiscano sull'analisi di quella vecchia non ancora completata.
Questo programma per il TI

99/4A permette di inserire le coordinate della prima generazione di organismi, seguendo le semplicissime istruzioni che ci darà il nostro computer. Si dovrà indicare prima la COLONNA (lettere A-P), poi di seguito si dovrà precisare la RIGA (numeri 1-16), ad esempio "H12", e quindi si dovrà premere ENTER.
Qualora si desideri correggere delle coordinate già impostate, è possibile farlo — purché, ovviamente, non sia stato già dato l'ENTER — utilizzando il tasto FCTN-S.
Comunque, una procedura di controllo inserita nel programma impedirà di compromettere l'esecuzione dell'algoritmo a causa dell'eventuale inserzione di coordinate non consentite (ad esempio T12 o F32), permettendo al computer di riconoscerle e di non accettarle. Se, però, vengono impostate delle coordinate incoerenti (ad esempio 5G6 o AC3) l'esecuzione del programma si interrompe inesorabilmente e sarà necessario dare nuovamente il RUN, ricominciando tutto daccapo.
Poiché si tratta di un gioco molto particolare, e certamente non di un "arcade game", la velocità non sarà una caratteristica necessaria del programma.
I tempi di inserzione delle coordinate degli organismi della prima generazione permettono di riflettere sulle successive coordinate che si intende inserire al fine di realizzare la configurazione di partenza ritenuta più interessante.
Inoltre, le successive generazioni appariranno soltanto dopo che il computer avrà terminato l'analisi delle immediate adiacenze di tutti i punti del no-

10 - 310	Presentazione
320 - 630	Dimensionamento matrici, definizione dei caratteri, stampa del campo della simulazione
640 - 980	Routine per l'inserzione delle coordinate degli organismi della prima generazione, controllo delle coordinate inserite e visualizzate degli organismi inseriti
990 - 1180	Routine per l'analisi delle adiacenze delle caselle e costruzione della generazione seguente
1190 - 1270	Stampa della nuova generazione
1280 - 1320	Costruzione di una nuova matrice per conservare la vecchia generazione
1330 - 1430	Routine che controlla se c'è almeno un organismo vivo ed invia l'apposito messaggio in caso negativo, con conseguente termine dell'esecuzione
1440 - 1500	Subroutine DISPLAY AT
1510 - 1800	Subroutine ACCEPT AT

Principali Variabili

A$	Contiene i caratteri che vengono stampati sullo schermo tramite la subroutine PRINT AT
NG	Numero progressivo delle generazioni
//Q// ed //U//	Matrici che contengono le varie generazioni
AA$	Contiene il valore della colonna degli organismi della prima generazione inseriti tramite tastiera
FF	Contiene il valore della colonna degli organismi della prima generazione inseriti tramite tastiera
G	Contiene il valore, analogo al precedente, della riga
VV	Contiene il numero di organismi vivi presenti nelle adiacenze della casella analizzata

stro spazio ed attuato una procedura di controllo per assicurarsi che vi sia ancora almeno un organismo vivo, permettendo nel frattempo di considerare la generazione visualizzata prima che venga sostituita dalla successiva, senza costringere ad una frenetica rincorsa visiva delle generazioni, che riuscirrebbe piuttosto fastidiosa, specialmente in caso di configurazioni molto complesse.
Ogni generazione viene individuata opportunamente con il suo numero d'ordine ed ogni volta che viene visualizzato un organismo, sia della prima generazione, sia delle successive, il computer emette un suono. Se non vi sono più organismi vivi appare un messaggio sullo schermo e termina la esecuzione del programma.
Altre caratteristiche del programma si possono facilmente rilevare durante l'esecuzione.
Al riguardo, è interessante notare che l'evoluzione di tutti i sistemi di organismi, anche dei più complessi, tende immancabilmente a ricondursi a tre risultati finali:
1) il sistema diventa periodico, ossia ritorna all'infinito a configurazioni precedenti, passando per un certo numero di generazioni intermedie, come quello della figura 1;
2) il sistema diventa stabile, ossia le generazioni, a partire da una certa configurazione, si ripetono uguali all'infinito, come quello della figura 2;
3) muoiono tutti gli organismi.
In tutti e tre i casi si fa apprezzare la possibilità di leggere al centro dello schermo, sotto il campo ove appaiono gli organismi, il numero progressivo della generazione nella quale si è, in un certo senso, esaurita l'entropia del sistema, che non può più evolversi, perché è diventato periodico.
Bene, non resta che rimandare ogni altra riflessione a dopo l'esecuzione del programma, che si presenta quasi sempre interessante, sia con configurazioni iniziali particolari, ad esempio variamente simmetriche, sia con configurazioni di partenza realizzate del tutto casualmente. ■

TI 99 NEWSOFT *ha compiuto un anno.*
Quale modo migliore di augurargli lunga vita
se non quello di abbonarsi?

Con sole L. 95.000 riceverete 11 numeri
della rivista e in più avrete diritto agli sconti
sul software e hardware offerto da **comPIUmarket.**

Per abbonarsi basta inviare un assegno
non trasferibile o un vaglia postale a
Downloaded from www.ti99iuc.it
NEWSOFT snc
Via S. Jacini, 4 - 20121 Milano

POLITICANZA

di SESTO AVOLIO

In tempi in cui si fa un gran parlare di intelligenza artificiale questo programma può essere detto di "intellighenzia" artificiale.
In realtà questo non è un gioco vero e proprio, non è un arcade, né un adventure, né un programma di simulazione; è soprattutto un "divertissement" sul tema delle discussioni politiche, ispirato anche dalle elezioni del 12 maggio scorso.
Immaginatevi una tavola rotonda fra 8 partiti politici, con un moderatore particolarmente abile, per cui nessun rappresentante di partito interrompe gli altri, ma aspetta pazientemente il proprio turno, ascoltando, o meglio, leggendo il parere degli avversari. A voi decidere a quale affermazione, a quale "linea" politica dare la fiducia. Come tutti ben sanno è piuttosto difficile, praticamente impossibile interrompere i politici; pertanto il programma disabilita l'uso del CLEAR (FCTN 4) e l'unico modo per farli smettere è il QUIT (FCTN =).

Descrizione del programma

Ogni asserzione può essere divisa in 7 parti: soggetto, predicato, complemento e 4 circostanze accessorie. Per ognuna di queste parti sono previste 10 possibilità, per un totale di 10.000.000 asserzioni sintatticamente corrette, anche se non necessariamente comprensibili, o di senso compiuto, d'altra parte, anche i politici, si sa...
Gran parte del programma è costituito dalle istruzioni DATA con le parti di asserzioni (sotto forma di stringhe).
Per ovviare alla mancanza di un RESTORE calcolato, tali stringhe sono lette e memorizzate sotto forma di un array 10 x 7, da cui, per estrazione casuale, viene prelevata l'asserzione (array 1 x 7).
Tale asserzione viene visualizzata sullo schermo carattere per carattere, simulando un procedimento tipo telescrivente. Per evidenziare il nome del partito che ha la parola si utilizzano degli SPRITES in MAGNIFY (2). È questa in pratica l'unica particolarità grafica del programma.
Alla fine di ogni giro di consultazione dei partiti il computer chiede se si vuol continuare o meno, ma è solo uno scherzo, infatti rimanda comunque ad un nuovo giro.
Come se non bastasse un UNBREAK alla riga 250 rende impossibile interrompere il programma.

■

TILANDIA

di PAOLO TAMBURELLI

Il gioco è la versione per il TI-99 del conosciutissimo *HAMMURABI,* una delle più complete simulazioni di governo.
Il gioco è ambientato nel medioevo; il giocatore si trova a dover sostituire il Re Astolfo, partito per le crociate, nella difficile conduzione del piccolo stato. Molti saranno gli ostacoli che si porranno davanti al giovane vassallo, il quale dovrà ingrandire il regno, nutrire i sudditi, arricchire il t esoro e, ogni 4 anni render conto del suo operato al Re, il quale se insoddisfatto, potrebbe degradarlo a stalliere. Ma non sarà solo il vostro sovrano a giudicarvi, anche il popolo non scherza e, se diventa cattivo non si limita a degradarvi, vi impicca.

Istruzioni per il gioco
La prima domanda che vi porrà il calcolatore sarà ACRI DA SEMINARE? Ovviamente dovrete rispondere con un valore massimo uguale a quello del vostro territorio seminabile, proporzionale alla manodopera e al grano che avete a disposizione per la semina (questi valori sono visualizzati sullo schermo). Dalla seconda alla quinta domanda vi vengono offerte le varie opportunità di vendita e/o acquisto con pagamento in grano o in scudi. La sesta e la settima riguardano le tasse che imponete ai vostri sudditi e alle quali dovrete porre particolare attenzione in quanto più sono alte, meno sono gradite. L'ottava vi chiede quanto intendete spendere per proteggere il grano dei vostri depositi dai famelici topi medioevali, noti per la loro voracità.
Ovviamente la sicurezza è proporzionale alla spesa. La nona domanda contempla la quantità di grano che intendete dare al popolo. Ricordatevi che la felicità, la prolificità, ed i morti di fame dipendono da questo fattore. La decima domanda infine, rappresenta le tasse che il sovrano legittimo (Re Astolfo) pretende da voi, il cui importo è riportato sul video. Se i vostri fondi sono insufficienti il sovrano vi espropierà tanti acri di terreno, quanti sono gli scudi mancanti. Dato che il gioco, specialmente per i più bravi può protrarsi parecchio, c'è la possibilità di interrompere la partita salvando su disco o nastro i risultati ottenuti, e riprenderla in seguito. Queste operazioni sono chiaramente spiegate dal calcolatore, è quindi sufficiente seguire le istruzioni che compariranno sul video.

■

Descrizione del listato

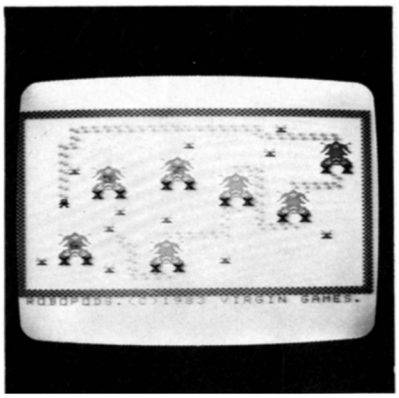
Lin. N.	
10 - 150	Pagina grafica di presentazione
160 - 220	Musichetta
230 - 250	Preparazione schermo
270 - 350	Istruzioni? / Inserimento nome / Anno di inizio
360 - 550	Presentazione e eventuale ripresa di vecchia partita
560 - 730	Condizioni iniziali
780 - 1520	Routines domande e calcoli loro relativi
1530 - 2180	Calcoli di fine semestre e calcolo del tempo
2190 - 2220	Decide eventuale guerra
2230 - 2300	Tabella principale
2310 - 2320	Linea di ritardo
2330 - 2460	Pagella ottenuta nel quadriennio
2470 - 2480	Linea di ritardo
2490 - 2510	Calcolo esito
2520 - 2930	Promozione ottenuta ed eventuale vincita
2940 - 3020	Routine di impiccagione
3030 - 3220	Routine guerra
3230 - 3510	Calcolo esito guerra
3520 - 3600	Stampa l'esito
3610 - 3740	Azzeramento variabili usate
3750 - 3850	Routine arruolamento mercenari
3860 - 3960	Routine lettura dati
3970 - 4060	Routine scrittura dati
4070 - 4270	Routine di print at ed input at
4280 - 4800	Routine di aggiornamento tabella

BUONE VACANZE

A TUTTI
I LETTORI DI
TI 99 NEWSOFT

TI99IUC.it

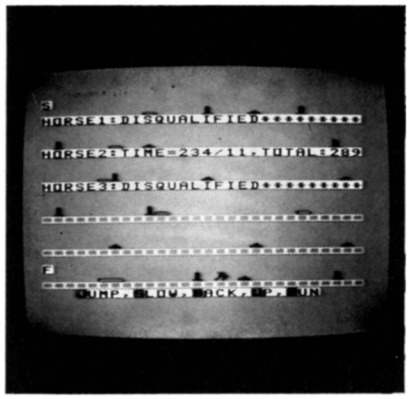
ROBOPODS



La peggiore invasione aliena che la terra abbia mai visto. Sulle città e i campi del pianeta i Robopods arrivano teletrasportati dalla astronave madre, e implacabilmente minano il terreno intorno a loro.
Sono mine potenti che al solo tocco di un piede saltano per aria. Ma non si limitano a piazzare delle mine, dovrete fare i conti con un pericolo ben più temibile: la radioattività.
Tutta la zona su cui vi muovete è stata contaminata da una sostanza mortale che viene attivata al vostro passaggio ogni volta che ci camminate sopra. Ad ogni passo lasciate dietro di voi la morte e non potrete tornare indietro a meno che non riusciate a rubare ai Robopods, le stuoie di sicurezza che dovrete buttare sopra le impronte.
Ma come fermarli?
I Robopods hanno un punto debole che li disattivizza.
Scopritelo e sarete salvi. Ma... dovrete disattivarli in sequenza altrimenti...
Questo appassionante gioco di invasione porta la firma di Malcom Adams, un ingegnere della programmazione sui piccoli computer. È disponibile nel catalogo COMPIUMARKET in fondo alla rivista.

FUN-PAC

In questa cassetta, distribuita dalla Virgin per il TI/99 ci sono due giochi: SUPERMOUSE e LET'S GO TO THE RACES.
Nel primo, SUPERMOUSE, protagonisti del gioco sono gli eterni nemici: il gatto e il topo e la loro sempiterna lotta.
Il topo, questa volta, ha però una possibilità di rivalsa: se infatti mangerà tutti i pezzetti di formaggio sparsi per il labirinto, si trasformerà in Supermouse e nei suoi panni si trasformerà da cacciato in cacciatore.
Nel secondo, invece, LET'S GO TO THE RACES, protagonista sarà la vostra abilità di cavallerizzo. È una gara campestre con ostacoli e pozze d'acqua da saltare e una schiera di avversari che frusteranno a più non posso i loro cavalli.



Entrambi i giochi sono disegnati da Karl Brenner, un programmatore austriaco che ama programmare giochi per piccoli computer nei momenti liberi del lavoro.
L'immissione di questo programma sul mercato è rilevante per i numerosissimi possessori di questo tipo di computer, la cui produzione è cessata in modo inaspettato.
Per riceverlo a casa richiedetelo compilando la cartolina del Compiumarket. ■

TI 99 CALC

Un versatile tabellone elettronico che vi permette di calcolare rapidamente il vostro bilancio annuale, le spese condominiali, di visualizzare istogrammi e moltissime altre applicazioni.

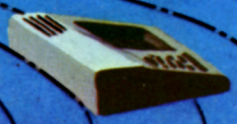
Il programma gira in Extended Basic e viene fornito su cassetta completo di un dettagliato manuale di istruzioni e di 4 esempi applicativi.

Richiedetelo, utilizzando l'apposita cartolina, al prezzo di Lit. 39.000 (35.000 per i nostri abbonati).

TI 99 NEWSOFT

L. 10.000

Numero 12 - anno II° - settembre 1985



COME FARE LE PULIZIE D'AUTUNNO AL VOSTRO COMPUTER

IMPARATE A SUONARE LA CHITARRA COL TI-99

IN SOFTWARE & SOFTWARE
Star Math - Mistery Tower
Duello - Planets Hunter
Strange Castle

TI99IUC.it

SOMMARIO

☐ **TI BASIC**
Bug e Debug pag. 3

☐ **PROGRAMMAZIONE**
Sub logica e Sub data pag. 4

☐ **TI EXTENDED**
Salvare capre e cavoli pag. 6

☐ **HARDWARE**
Le pulizie di ...settembre pag. 7

☐ **TI EXTENDED + 32K**
Due routine in L/M pag. 10

☐ **STATISTICA**
Risoluzioni di equazioni di II e III grado pag. 11

☐ **TI-99/4A**
L'unità a cassette pag. 13

☐ **TI-STRANGE CASTLE**
Alla caccia dei gioielli rubati pag. 14

☐ **COMPUTERNEWS**
Novità da tutto il mondo pag. 15

☐ **GLI ACCORDI ALLA CHITARRA**
Un omaggio di "TI-99 Newsoft"
all'Anno Internazionale della Musica pag. 18

☐ **DUELLO**
Torniamo nel vecchio far-west pag. 20

☐ **MISTER TOWER**
Una fiaba autunnale pag. 21

☐ **PLANETS HUNTER**
Avventura tra le galassie pag. 22

☐ **STAR MATH**
Una utility per imparare la matematica elementare pag. 24

☐ **LA VIGNETTA** pag. 25

☐ **LA POSTA** pag. 26

☐ **PICCOLI ANNUNCI** pag. 27

☐ **COMPIUMARKET** pag. 28

LATO 1	LATO 2
TI BASIC:	**EXTENDED BASIC:**
☐ Mistery tower	☐ TI strange castle (2 programmi)
☐ Teach bit	☐ Star math
☐ Duello	☐ Planets hunter (2 programmi)
☐ Equazioni di II grado	☐ Accordi alla chitarra
☐ Equazioni di III grado	
EXTENDED + 32K	
☐ CSAVE CLOAD	

TI 99 NEWSOFT

ANNO II° - Numero DODICI
SETTEMBRE 1985
MENSILE
Direttore responsabile:
ANTONIO C. LOSITO
Coordinatore Editoriale:
VINICIO CIVININI
Direttore Amministrativo:
ENRICO CALLERIO
Art Director:
BEPPE RE-FRASCHINI
Illustrazioni:
MARIA TERESA COCOZZA
Segreteria:
CRISTINA FRISON
Redazione:
FIORENZA AURIEMMA
Collaboratori:
SERGIO BORSANI - GIANPAOLO BOTTIN - DANIELE CATALFAMO - PAOLO CIVARDI - STEFANO DELLI PONTI - FRANCESCO MACALI - MAURO MATTIAZZI - EZIO MONTINI - LEVIO PEROTTI - RICCARDO ROSSI - MARCO SQUINTANI - ALBERTO STRAFILE - SEBASTIANO TOMASELLO
TI 99 NEWSOFT è una pubblicazione della **NEWSOFT snc.**
Via Stefano Jacini, 4 - 20121 MI
Tel. 02/807464 - Telex 324284
REGISTRAZIONE AL TRIBUNALE DI MILANO N. 323 - DEL 7.7.84

© COPYRIGHT NEWSOFT snc
TUTTI I DIRITTI RISERVATI
Manoscritti e foto originali, anche se non pubblicati, non si restituiscono ed è vietata la riproduzione anche parziale di testi, fotografie e listati.

Pubblicità:
GERMANO DIVERIO
Tel. 02/862534
Abbonamenti: l'abbonamento annuo a 11 numeri di **TI 99 NEWSOFT** costa L. 95.000 per l'Italia; L. 160.000 per l'Europa (spedizione aerea); L. 200.000 per tutti gli altri paesi. L'importo, per assegno non trasferibile o per vaglia postale, va inviato a **NEWSOFT snc,** Via S. Jacini, 4 - 20121 Milano
Stampa: LALITOTIPO srl - Milano
Concessionaria per la distrib.: EUROSTAMPA Srl - C.so Vittorio Emanuele, 11 - 10123 Torino
Telefono 011/538166-7

L'editore non si assume responsabilità alcuna sul contenuto degli articoli e/o software inviati. Per gli articoli firmati e/o siglati da collaboratori esterni, la redazione si assume la responsabilità prevista dalle leggi sulla stampa.

Tutti possono collaborare a TI 99 NEWSOFT, sia con articoli che con programmi. Scriveteci o telefonateci in redazione per avere tutti i ragguagli.

Downloaded from www.ti99iuc.it

BUG E DEBUG

di MARCO SQUINTANI

Chi non ha mai sentito parlare di debugging? Ben pochi, crediamo, almeno tra i più affezionati lettori della rivista. Comunque sia, per una eventuale, ristretta minoranza, affrontiamo il problema alla larga e portiamo il discorso alle origini per meglio comprendere l'inconveniente.

Innanzitutto, il ''bug'' (o pulce, in italiano) è quel subdolo errorino di calcolo, di sintassi e di logica che manda il computer letteralmente ''nel pallone'' facendogli emettere messaggi di errore a raffica. Il TI BASIC, a questo proposito, ci dà tuttavia una mano indicandoci il più delle volte il numero di linea in cui la nostra abilità di programmatore ha fatto un passo falso. Alcuni errori vengono segnalati dal computer subito dopo l'immissione della linea (quando si preme ENTER, insomma) e questo fatto, purtroppo, provoca la perdita irreparabile della linea stessa, che andrà riscritta correttamente dall'inizio (in EXTENDED, invece, esiste l'unico comando di REDO o FCTN 8 che recupera il tutto senza danno). La rettifica di questo tipo di sbaglio, anche se noiosa, non comporta comunque grandi difficoltà di ordine pratico.

Effettuato questo primo sommario controllo, il nostro computer passerà ad una analisi più accurata durante la generazione della tavola dei simboli. Senza entrare nel difficile, diciamo che il TI 99 crea un'area di memoria nella quale pone, memorizzandoli, variabili, matrici, funzioni, ecc. Scovare la causa dell'interruzione del programma a questo punto è già problematico, specie se ci si imbatte in messaggi del genere FOR NEXT ERROR o un INCORRECT STATEMENT che riguardi le matrici (istruzioni con le DIM) e quindi anche NAME CONFLICT.

Passato al meglio questo secondo scoglio, il programma deve ancora sottostare all'occhio vigile del terzo ed ultimo controllo operato su tutte le sue parti. Si è già all'interno dell'esecuzione del programma e il calcolatore segnala tutte quelle istruzioni che - poveri noi - non riesce ad eseguire per vari motivi. Anche qui la stampa dei messaggi può causare il momentaneo ''blocco'' del computer o, alla meno peggio, mettervi solo al corrente della brutta situazione.

Grande pregio del calcolatore in queste situazioni è quello di mantenere i valori di tutte le variabili (numeriche, di stringa, di matrice, ecc.) inalterati, come prima della segnalazione di errore. E allora cosa ce ne cala (per usare un eufemismo)? Se avrete la costanza di seguirci anche nella prossima puntata vedrete appagate queste e altre domande (legittime del resto) su come comportarsi per fare finalmente girare senza intoppi quel dannatissimo programma, causa delle vostre (e anche nostre) insonnie. A presto.

■

ADVENTURLAND SUB LOGICA e SUB DATA

di DANIELE CATALFAMO

Siamo così arrivati, quasi senza accorgercene, alla terza puntata di ADVENTURLAND. Questa volta, finiremo di esaminare la struttura generale che dovrà possedere il nostro Adventure, completeremo le SUB ROUTINE, - iniziate le volte scorse -, daremo la soluzione dell'avventura presentata sullo scorso numero - TI PUFFI - e ne proporremo uno nuovo.
Colgo l'occasione per ricordarvi di non perdere il prossimo numero di "TI 99 New Soft" su cui troverete un'avventura notevolissima, contenente tutto ciò che deve avere un programma degno di questo nome: SALVATAGGIO / CARICAMENTO della situazione corrente, VARIETÀ di messaggi, REBUS risolvere,... Completiamo ora la descrizione dei due sottoprogrammi rimastici dalla scorsa volta:
SUB LOGICA e SUB DATA.

Sub Logica
Questo è probabilmente il più complicato di tutti gli altri da implementare sul calcolatore. Deve infatti prevedere ogni tipo di situazine che si può verificare e prevedere ciò che farà l'utente. Dovrà inoltre evitare che succedano cose strane. Mi spiego meglio. Se ci troviamo in una grande pianura con un grosso albero al centro, l'utente non potrà prendersi la grande pianta e portarsela a spasso per tutta la durata dell'avventura, - a meno che non sia Maciste o Ercole. Oppure se vi è una porta chiusa, il programma dovrà tenere conto di questo e lasciare passare il nostro eroe, solo se questi riuscirà ad aprire la porta. Inoltre bisognerà controllare la validità del comando impartito dall'utente e, se valido, farlo eseguire.
Come avrete capito, ci sono non poche difficoltà. Per ovviare a tutti questi incovenienti ci sono vari modi, noi ne proporremo uno, a voi la decisione di usarlo, modificarlo, migliorandolo o scartarlo del tutto.
Tutto deriva, direttamente o indirettamente, dal comando impartito dall'utente. La prima cosa che faremo sarà allora di controllare se sia il VERBO che l'OGGETTO si trovano nelle rispetive tabelle. In caso negativo non faremo altro che passare il controllo al SUB MESSAGGI, il quale segnalerà l'errore verificatosi. Se, invece, sia il verbo che l'oggetto sono presenti nelle tabelle, dovremo verificare innanzi tutto la validità del comando. A tale scopo, ad ogni verbo abbineremo tutti gli oggetti insieme ai quali il comando abbia un senso logico, in caso contrario passeremo il controllo, come prima al SUB MESSAGGI. Ogni verbo avrà quindi associati gli oggetti che gli forniscono un senso logico e, ad ognuno, corrisponderà

una certa azione, conforme al comando dato. Per quanto riguarda poi tutte le situazioni particolari che potrebbero venirsi a creare, non resta altro da fare che assegnare a delle variabili dei valori stabiliti, i quali saranno codificati dal programma in maniera opportuna. Ad esempio nei programmi che vi ho proposto fino ad ora, e in quelli che seguiranno, ho una variabile per ogni oggetto dell'avventura. Questa variabile numerica, contiene il numero della stanza in cui l'oggetto si trova. Così facendo quando stamperò la descrizione della stanza, non dovrò fare altro che controllare tutti gli oggetti che hanno lo stesso numero e stamparli con essa. Se poi l'oggetto viene preso dal giocatore metto [—1] nella variabile. Se, ancora, l'oggetto dovrà essere trovato in qualche maniera, - può essere sottoterra come la PATATE, - allora nella variabile ci sarà [0].

Sub Data

Questo blocco è, insieme al blocco LOGICA, la parte più importante di tutto il programma. In esso sono contenuti, il vocabolario, gli oggetti con la relativa descrizione, tutti i commenti che il blocco MESSAGGI leggerà e stamperà e le descrizioni di tutte e quante le stanze.

L'unico problema è creato dalle tabelle. Noi abbiamo scelto due soluzioni. La prima, - più semplice - memorizza il vocabolario, cioè tutti i verbi compresi dal calcolatore, e tutti gli oggetti in due matrici di stringa. Per controllare che il verbo e l'oggetto siano nelle tabelle basterà controllare se si trovano nelle matrici. Unico appunto da fare: il dispendio di memoria è eccessivo, e quando si hanno ''14 Kbyte...''.

Certo si potrebbe anche pensare di non RIEMPIRE le matrici con dei dati residenti nel programma stesso ma con dei FILES caricati dal registratore. Purtroppo l'eccessiva lentezza del registratore a cassette ha fatto scartare quasi immediatamente questa variazione.

Il secondo metodo, invece, non fa ricorso ad alcuna matrice. Si limita semplicemente a controllare direttamente nelle linee DATA, con opportune e calcolate READ e RESTORE. È ovvio allora capire la nostra scelta verso una soluzione molto più complicata, ma convenientissima dal punto di RAM usata.

Anche per questo numero, la parte dedicata alla costruzione del nostro modello è terminata. Come sempre rinnovo l'invito a tutti coloro che avessero risolto almeno uno degli ADVENTURE TEXAS a scriverci, ma soprattutto imploro quelli che possiedono la cartuccia *Return to pirate's isle* a comunicarci dove sono arrivati, - e come!!!, Noi non riusciamo più a muoverci.

Ringraziando tutti do appuntamento, oltre che al prossimo numero, alla soluzione di TI PUFFI, ed alla nuova avventura: STRANGE CASTLE.

Buon Divertimento!

■

È uscito il primo numero di

SOFTWARE & SOFTWARE

Downloaded from www.ti99iuc.it

una nuova rivista dedicata al TI-99 4/A, dove troverete i listati di alcuni tra i migliori programmi per il vostro computer

RICHIEDETELA IN EDICOLA COSTA SOLO Lire 3.000

SALVARE CAPRA E CAVOLI

di MARCO SQUINTANI

Dopo esserci dilungati a lungo sulle sprites e sul loro utilizzo migliore, apriamo ancora una parentesi per illustrarvi una piccola routine dalle grandi possibilità.
La sua scoperta è da attribuire ad un nostro acuto lettore. *Patrizio Scarsi,* che ce ne ha gentilmente fornito la chiave di lettura. Il programmino, pur nella sua esiguità (due righe appena), fornisce un aiuto non trascurabile per chiunque intenda passare dal sistema operativo Extended Basic a quello Ti Basic, senza voler perdere il programma residente in memoria. Come chiaro esempio si pensi a quando erroneamente si caricava in ambiente Extended un programma che girava solo in TI Basic, dovendo così spegnere tutto e ripetere da capo le operazioni di caricamento. La faccenda ora si può risolvere a questo modo

```
1! @ P- 2 CALL COLOR (0,1,1)
```

Per trasferire il nostro programma in memoria da un linguaggio all'altro basterà dunque inserire all'inizio dello stesso queste due righe, usando alcuni accorgimenti. Non si dovrà infatti cambiare l'ordine di esecuzione del programmino e non compattarlo sfruttando i doppi due punti dell'Extended. Una volta inserito e attivato il tutto con il sempiterno RUN apparirà inspiegabilmente il messaggio d'errore BAD VALUE IN 2 dettato dall'interprete del Ti Basic che non ammette il valore zero nella CALL COLOR.
Ora che si è finalmente passati al Ti Basic si dovranno eliminare le prime linee che, oltretutto, causerebbero problemi di "traduzione" facendo stampare strani caratteri e simboli qualora si voglia vederne l'aspetto con un LIST o con un EDIT.
Via il dente, via il dolore di estenuanti caricamenti "a vuoto". Da questo momento il programma residente potrà essere manipolato a piacere, come di consueto, eseguito, listato, risalvato ecc. ecc.
Attenzione, però: il passaggio è irreversibile, non si può infatti passare con lo stesso "trucchetto" dal Ti Basic al Ti Extended. Questa routine è aperta ad ogni altro eventuale tipo d'impiego; preghiamo quindi i nostri lettori di comunicarci a tempo le loro scoperte.

■

PULIZIE DI SETTEMBRE

di DANIELE FIGURATI

Vi è mai capitato che lavorando (o giocando) col TI-99 e usando un modulo S.S.S., magari l'onnipresente Extended Basic, il video si riempia di caratteri senza senso, o il programma si blocchi, o un suono cupo vi raggeli al pensiero di una prematura fine del vostro amico a sedici bit?
Ebbene, d'ora in poi, in questi casi, non occorrerà più spedire il tutto a Cittaducale: vi spieghiamo, di seguito, come fare per ridare la possibilità al volenteroso TI 99 di poter di nuovo leggere le GROM contenute nei suoi moduli, semplicemente rimuovendo lo sporco dovuto alla polvere che si accumula nella contattiera degli S.S.S., e, con esso, tutta la serie dei difetti suddetti.

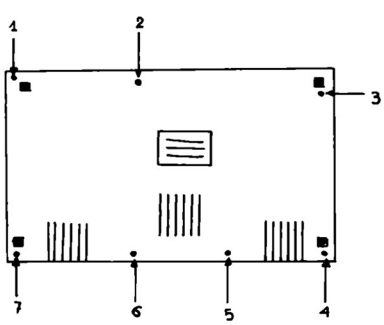


fig. 1

L'operazione, come vedrete, non è immediata, data la posizione inaccessibile di tale contattiera, ma, se ci seguirete con la dovuta attenzione, il vostro TI malato tornerà a nuova vita.
Staccate quindi tutti i cavi collegati alla consolle, portatevi il paziente su un tavolo sgombro e siate pronti per iniziare l'intervento.
Cominciate con il capovolgere il TI mettendolo con la tastiera verso di voi, e con un cacciavite a stella di media grandezza svitate le sette viti poste sul fondo (fig. 1). Esistono varie versioni, dal punto di vista dell'assemblaggio, del nostro computer, che si distinguono tra loro per alcune scelte costruttive, comunque di minima importanza. Ad esempio, mentre in una versione le sette viti appena tolte sono tra loro identiche può accadere che nella vostra ciò non sia vero, in tal caso il nostro consiglio è di annotarvene le rispettive posizioni su un foglietto, in quanto incontreremo ancora altre viti.

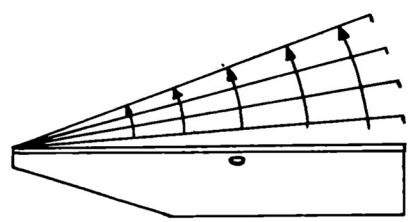


fig. 2

Sollevate ora il fondo del TI con un movimento a cerniera dal dietro in avanti (figura 2). Tale movimento disimpegnerà con uno scatto, la parte esterna dell'interruttore di alimentazione. Recuperatela e riponetela con il fondo e lo sportellino scorrevole situato sulla destra, che estrarrete dalle sue guide semplicemente sollevandolo,

TI99IUC.it

in un posto sicuro, insieme alle sette viti già tolte.
Vi appariranno ora la basetta dell'alimentazione, la parte inferiore della tastiera e, coperta dalla sua dispendiosa schermatura, la piastra madre (figura 3).
Per poter procedere con lo smontaggio dovete ora "attaccare" per prima la basetta dell'alimentazione. Essa si presenta a voi dal lato delle saldature, impedendovi, quindi, la vista della spina a quattro pins che la collega alla piastra madre. Senza scoraggiarvi per questa prima difficoltà cominciate col togliere le due viti poste agli angoli destri della basetta e scalzatela delicatamente dagli incastri che la trattengono. Liberate ora il filo (in realtà sono tre: bianco, rosso e nero, in quasi tutte le versioni) di collegamento con il trasformatore esterno, estraendo l'attacco dalla sua sede sulla parte posteriore dell'apparecchio, ed ora, sporgendovi un po' in avanti, potrete vedere chiaramente lo spinotto, l'unico in cui entrano quattro fili (in genere marroni) saldamente attaccato alla basetta dal lato componenti (figura 4).
Affacciandovi dalla parte dei componenti, vedrete che, rivolto verso l'interno della basetta, questo spinotto porta un fermo che bisogna disimpegnare premendolo all'estremità opposta (figura 5). L'operazione, di per sè complicata per un profano, è inoltre ostacolata anche dalla presenza di un grosso condensatore elettrolitico, che impedisce la facile presa della pinzetta da sbloccare. Il tutto è, comunque, abbastanza solido ed i vostri decisi ma non violenti tentativi non dovrebbero provocare alcun danno.
Liberato lo spinotto, la scheda dell'alimentazione sarà vostra preda e, guardandola, potrete gustare un saggio dell'ordine, della precisione e della bontà dei componenti usati dalla Texas nella fabbricazione di questo incompreso computer.
L'interessante descrizione dei singoli componenti hard la sacrifichiamo e passiamo all'assalto della piastra madre, notando prima, però, che in corrispondenza dell'interuttore di alimentazione è rimasta, nel contenitore, la parte meccanica interna in plastica di questo. Potete non toglierla dalla sua sede, ma se lo fate, prima di riporla cercate di memorizzare perfettamente il modo in cui è alloggiata sulla guida singola su cui scorre.
Le viti che trattengono la piastra madre sono tre. Svitatele, disincastrate la piastra muovendola leggermente, poi sfilate verso il basso il connettore della tastiera, situato al di sotto, con un movimento delicato, onde non piegare i sottili pins di collegamento, ed estraete la piastra (figura 6).
Capovolgendola ci appare, a destra, il connettore incriminato. Estraetelo dalla sua sede prendendolo per i bordi del cir-

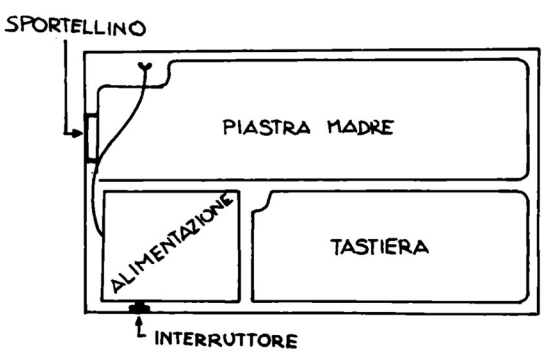


fig. 3

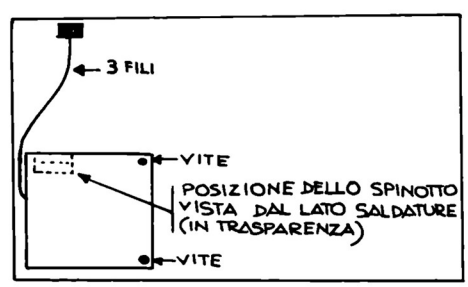


fig. 4

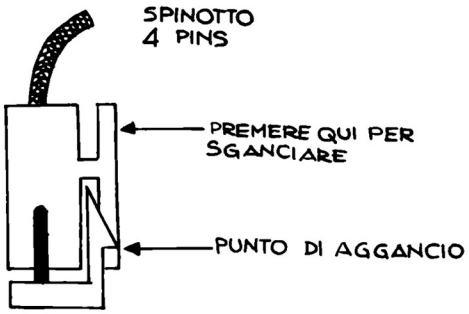


fig. 5

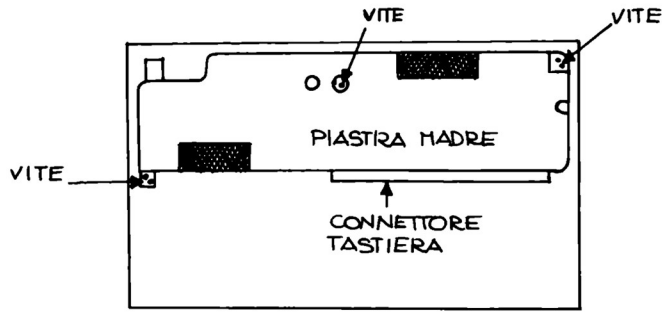


fig. 6

cuito stampato su cui è alloggiato, facendo forza verso l'alto. Riponete la piastra madre in un posto sicuro e prendete il connettore.
Rigirandolo un po' tra le mani, noterete che la contattiera interna è protetta da una mascherina nera con una feritoia. Per rimuoverla basta sbloccarla alle estremità con un cacciavitino o, addirittura, con un'unghia, essendo di una plastica molto duttile (figura 7).
A questo punto la contattiera è alla vostra mercè e vi potete sbizzarrire a pulirla con il metodo più opportuno che la vostra immaginazione suggerisce, facendo attenzione, però, a non forzare sui contatti che, seppure metallici, sono piuttosto delicati. Per i meno fantasiosi andrà comunque bene il collaudatissimo cotton-fiock e alcool dai risultati sicuri. Per quelli che vogliono fare le cose per bene, consigliamo di cercare di pulire i contatti in profondità. Noi ci siamo avvalsi di un fazzoletto molto sottile, spinto all'interno della contattiera grazie ad un chiodino d'acciaio, inutile dire che la delicatezza è d'obbligo e che l'alcool è indispensabile. Per rimontare il tutto, se avete seguito scrupolosamente queste istruzioni, non dovreste avere problemi ad eseguire di nuovo il tutto all'indietro.
Rimettere la mascherina al connettore e reinserirlo sulla piastra madre. Capovolgere la piastra madre, inserire il connettore della tastiera, rimetterla in sede e avvitare le tre viti senza forzarle.
Rimettere la parte meccanica interna dell'interruttore (se l'avete tolta), riattaccare lo spinotto alla basetta dell'alimentazione e posizionarla in modo che la levetta dell'interuttore saldatovi dal lato componenti finisca nel buco della parte meccanica interna e il LED rosso (luce spia) nell'apposita scanalatura, dopo di chè avvitare le due viti senza forzarle.
Rimettere nelle loro sedi la presa dell'alimentazione esterna e lo sportellino scorrevole sulla destra.
Posizionare il fondo prestando attenzione che vada perfetta

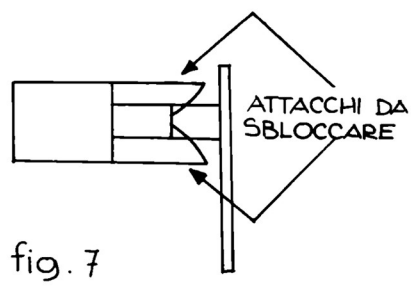


fig. 7

mente nella sua sede e avvitare le sette viti senza forzarle.
Inserire la parte esterna dell'interuttore nell'apposito buco fino allo scatto di bloccaggio.
Fatto!!! Ora il buon TI non dovrebbe avere ulteriori problemi relativi al funzionamento con gli S.S.S.
Sappiamo che a molti sarebbe piaciuto dare un'occhiata alla piastra madre, sbucciandola della sua schermatura, onde conoscere personalmente il misterioso TMS 9900. Attenti però a non prendere rischiose iniziative.

■

CSAVE - CLOAD (& GPLLNK)

di RICCARDO ROSSI

Questo mese abbiamo preparato per voi due routine in L/M che potranno esservi molto utili per lo sviluppo di programmi assembler in Extended Basic più espansione di memoria in quanto permettono di salvare e ricaricare, con il semplice registratore a cassetta, zone di memoria gestite dalla CPU.
Le due routine non sono certo idea originale ma hanno il pregio, rispetto a quelle incircolazione, di essere pronte all'uso e non necessitano di strane CALL LOAD per poter funzionare. Per poterle utilizzare dovete caricare e naturalmente far girare il programma che trovate sulla cassetta in Extended. Una volta caricate le due routine rimarranno in memoria fino a che non spegnerete il calcolatore o eseguirete una CALL INIT.
La prima delle due routine, come già detto, serve per salvare una zona di memoria ed è utilizzabile scrivendo:
CALL LINK ("CSAVE", INDIRIZZO, NUMBYTES)
in cui INDIRIZZO rappresenta l'indirizzo da cui partire per salvare la memoria. Questo indirizzo viene espresso come nel la CALL LOAD, in decimale come segue:
Indirizzo esadecimale
da > 0000 a > 7FFF
da > 8000 a > FFFF

Indirizzo decimale
da 0 a 32767
da -32768 a -1

NUMBYTES indica il numero di bytes da salvare per un massimo di 8192.
L'altra routine è utilizzabile digitando:
CALL LINK ("CLOAD")
e permette di ricaricare una zona di memoria precedentemente salvata con CSAVE.
La struttura del record (vedi figura 1) è la stessa utilizzata dai comandi "L" ed "S" dell'easy bug della mini memory e viceversa.
Questo non significa che i programmi scritti per la mini memory non funzionano in Extended Basic, ma che esiste la possibilità di utilizzare il programma assembler per l'extended.
La prima riguarda l'utility GPLLNK che è stata inserita per attivare il registratore. Se intendete utilizzare la GPLLNK il vostro programma assembler dovrà contenere le seguenti istruzioni:
GPLLNK EQU > 2500

BLWP GPLLNK
DATA XXXX
codice d'istruzione
Per quanto riguarda l'area di memoria disponibile nella low memory riservata all'assembler, questa va da 26A4 a 3FFO.
Nel momento in cui questi programmi vengono utilizzati, il contenuto della vdp viene distrutto. A chi volesse utilizzarli all'interno di un programma basic consigliamo di eseguire un RUN dopo le due call, in questo modo la vdp viene rimessa a posto e le variabili azzerate.
Esempio.

```
10 CALL LINK ("CLOAD")
20 RUN 30
30 REM IL PROGRAMMA
CONTINUA
```

■

RISOLUZIONE DI EQUAZIONI DI SECONDO E TERZO GRADO

di PAOLO CIVARDI

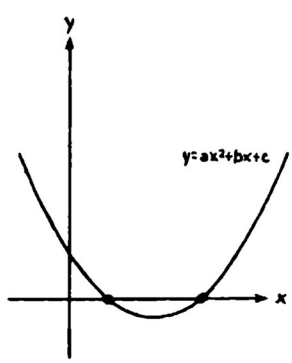
Nel corso della nostra succinta introduzione alla statistica abbiamo spesso parlato di correlazione tra causa-effetto, ovvero di fenomeni che per poter essere spiegati e descritti richiedevano l'impiego di equazioni lineari; abbastanza semplici nella costruzione e nella risoluzione. È però palese a tutti i nostri lettori che la Natura non è troppo spesso lineare nelle sue manifestazioni e sovente ci costringe a ideare moduli matematici più complessi nel tentativo di interpretare più fedelmente possibile quello che i nostri occhi vedono.

Tutti ricordiamo che a scuola, immediatamente dopo la risoluzione dell'equazione della retta, i nostri insegnanti di matematica introducevano le equazioni di secondo e di terzo grado: ebbene, è proprio quello che cercheremo oggi di fare eseguire al nostro computer che è un compagno fedele e attento ma ha bisogno di molte istruzioni (leggi righe di programma) per eseguire un'operazione abbastanza semplice; mentre il tentativo di far imparare la matematica ai nostri TI costituirà per noi un'occasione per rinfrescare la memoria.

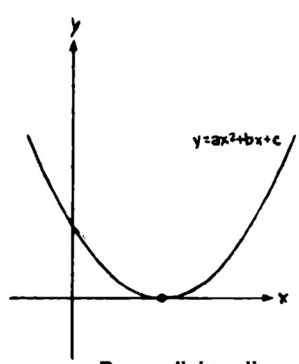
Risoluzione delle equazioni di secondo grado.

Una equazione di secondo grado ha la forma generale $ax^2 + bx + c = 0$ dove b e c possono assumere qualsiasi valore mentre a deve essere sempre diverso da 0, in caso contrario l'equazione diverrebbe di 1° grado.

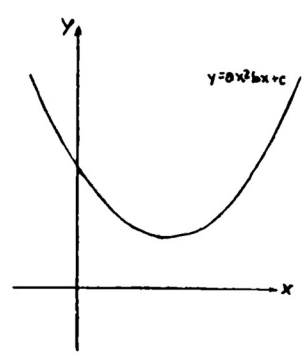
Il nostro calcolatore è in grado di trovare i valori di x che soddisfano un'equazione di questo tipo, e capace cioè di ricercare i punti in cui il grafico della funzione $y = ax^2 + bx + c$ interseca l'asse delle x. Dalle figure illustrate sotto notiamo che tre sono i casi possibili:



Due radici reali



Due radici reali coincidenti



Nessuna radice reale

La curva può intersecare l'asse delle x in due punti, in un solo punto, oppure non toccare mai l'asse. In questi casi avremo rispettivamente: 2 radici reali, due radici reali coincidenti oppure nessuna radice reale. (Il termine "reale" viene usato per avere una distinzione dalle radici complesse espresse da $\sqrt{-1} = i$).
Una parola circa il programma. La formula risolutiva della nostra equazione di 2° grado è

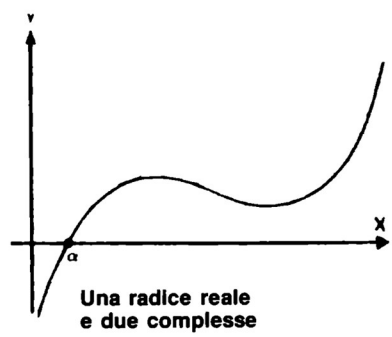
$$x = \frac{-b + \sqrt{(b^2 - 4ac}}{2a}$$

dove il termine $b^2 - 4ac$ viene chiamato "discriminante".
Dopo avere eseguito un controllo per assicurare che il valore di a sia diverso da 0, onde evitare una divisione con zero e provocare l'arresto del programma, viene eseguito il calcolo del discriminante.
Nel caso il risultato sia uguale a zero, le due radici saranno uguali e viene stampato il valore di esse.
Se il discriminante risulta negativo, l'equazione ha due radici complesse e viene stampato un messaggio di avvertimento.
Nel caso, infine, in cui il risultato sia positivo, vengono calcolate e stampate le due radici e quindi viene richiesto se si desideri eseguie un nuovo calcolo.
Per l'immissione dei dati è sufficiente battere i valori dei coefficienti a,b e c, separandoli con la virgola.

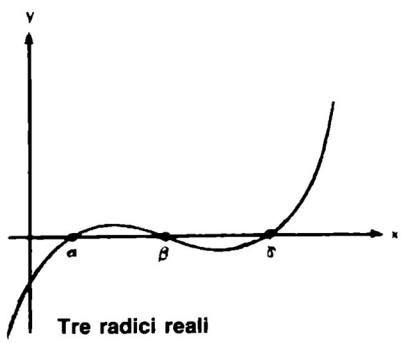
Risoluzione di equazioni cubiche

Per risolvere un'equazione di 3° grado avente la forma
$ax^3 + bx^2 + cx + d = 0$
è necessario trovare i luoghi in cui il grafico di
$y = ax^3\ bx^2 + cx + d$
intercetta l'asse delle x.

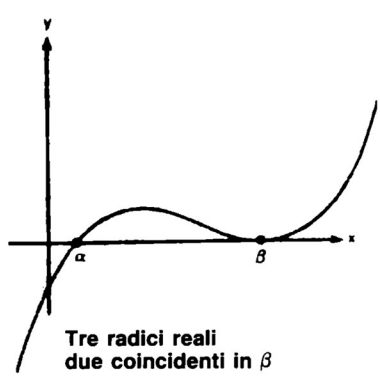
I casi possibili sono tre:



Una radice reale e due complesse



Tre radici reali



Tre radici reali due coincidenti in β

Anche in questo caso è necessario avviare il programma introducendo i valori dei coefficienti a,b,c, e separandoli con virgole.
Per il resto tutto si svolge come nella risoluzione delle equazioni di secondo grado.
Buon lavoro.

■

L'UNITÀ A CASSETTE

di MARCO SQUINTANI

Concluso il capitolo generale introduttivo, passiamo ora a suggerire piccoli trucchi e accorgimenti per migliorare i vostri rapporti di convivenza con l'ormai inseparabile registratore a cassette, meglio conosciuto come CS1.
Partiamo dalle cose fondamentali: pochi, all'atto dell'acquisto dell'indispensabile accessorio, si ricordano di dare anche un'occhiata al manuale del computer, tentando poi a più riprese di connettere da soli il tutto con risultati tipo l'inserzione del cavo d'interfaccia nella porta dei joystick o quello di invertire l'ordine di connessione dei tre cavetti col registratore.
Il capo a spina singola con il connettore a 9 pin va inserito sul lato posteriore del computer vicino alla presa dell'alimentatore; l'altro capo, a tre spine, va invece messo nelle prese (o jack) del registratore stesso. La spina con il filo rosso va inserita nella presa per il microfono (MIC), mentre quella con il filo bianco nella presa per la cuffia (EAR); quella con il filo nero, più sottile, infine, va nella presa dei controlli a distanza (REMOTE). Fatto questo, il problema più grosso è risolto.
Come seconda operazione dovrete mettere il tono degli acuti e quello del volume a poco più della metà; chi avesse solo l'indicatore del volume lo ponga invece al massimo. Cercate adesso di porre l'unità lontana dal televisore (per quanto lo consenta il cavo) e non allineato su una stessa superficie metallica con il computer.
È arrivato il momento di caricare i vostri dati. Se i disturbi continuano, provate a caricare più volte il programma variando il tono del volume e degli acuti, poiché tra i due registratori (quello usato per salvare i dati e quello utilizzato per leggerli) ci potrebbe essere incompatibilità come, per esempio, tra i registratori europei (Philips, Irradio, ecc.) e quelli giapponesi (Sony, Toshiba e via discorrendo).
Ormai sull'orlo della disperazione, vi accorgete che nonostante tutto il registratore è quanto mai restio alla lettura dei vostri sospirati programmi. Date allora un'occhiata alla testina, al cabestano e al Pinchroller (a destra della testina). Se sono un po' sporchini, datevi la pena di pulirli (la cosa va fatta almeno una volta ogni tre mesi, regolarmente), altrimenti cominciate pure a pensare al fattore "magnetizzazione". Questo subdolo inconveniente è prodotto dal nastro stesso che deposita, trascinandosi, particelle di ferrite un po' ovunque "inquinando", in caso di sovrannumero, la lettura dei dati. Se anche questo espediente desse risultato negativo, ecco gli ultimi disperati tentativi: cattiva velocità di trascinamento del nastro e inesatta taratura dell'azimuth. La prima situazioen è controllabile avendo a disposizione una cassetta stroboscopica, la seconda richiede invece una procedura particolare. Fate cioè emettere al registratore una nota o un suono costante e agite sulla vite regolabile posta a lato della testina e a cui si può accedere con un cacciavite sottile. Regolate la suddetta vite fino a che non sentirete il più chiaro possibile il suono in uscita, controllabile anche mediante gli indicatori di livello (VU-meter) o con uno strumento di misura esterno.
Se malgrado tutto ciò la situazione ancora non si sblocca, non preoccupatevi più di tanto: potrete sempre prendervela con il computer (o con noi)...
Sulla cassetta trovate comunque un semplice programmino per verificare il vostro livello di tecnici della registrazione.

TI STRANGE CASTLE

di DANIELE CATALFAMO

Come promesso la volta scorsa vi diamo la soluzione dell'avventura pubblicata sul numero precedente: TI PUFFI.
In tutto le stanze erano 36. Alcune di queste necessitavano di trucchi o di opportuni ragionamenti per poterci entrare. Vi erano comunque tre diverse possibili soluzioni. In ognuna di queste, però, era necessario trovare, svegliare e portarsi dietro proprio lui: IL GRANDE PUFFO.

Prima soluzione
Questa si proponeva di far valicare al nostro eroe i monti e giungere attraverso un passaggio segreto direttamente al castello del cattivissimo Gargamella.
Le azioni necessarie erano le seguenti:
— Prendere le SCARPE;
— Andare a prendere GRANDE PUFFO, nel deserto;
— Andare sui monti fino dai MORGS;
— Qui togliersi le SCARPE;
— Spaventare così i MORGS;
— Si apre allora una via ad EST che conduceva al castello;
— Bere la POZIONE così da svegliare GRANDE PUFFO;
— Andare da GARGAMELLA e chiamare GRANDE PUFFO;
— Si apre così un passaggio nella stanza 28;
— Andare nella stanza 35 e puffare la PUFFETTA.

Seconda soluzione
Questa conduceva al castello via acqua.
— Bisognava prendere GRANDE PUFFO nel deserto;
— Svegliarlo (Vedremo sotto COME!);
— Prendere la BARCA;
— Andare fino dalla BALENA;
— A questo punto chiamare GRANDE PUFFO e venire condotti proprio davanti alla porta del CASTELLO;
— Dopo analogo alla prima.

Terza soluzione
— tagliare la BARCA e prenderne un pezzo;
— Prendere il CAMMELLO;
— Percuotere il CAMMELLO e allora si sveglia GRANDE PUFFO;
— Baciando il CAMMELLO questi scappa in una grotta nascosta che conduce, (MANCO A DIRLO!!!) al castello;
— Dopo analogo alla prima.

Trabocchetti
I pericoli da affrontare non erano poi molti, e ve li elenchiamo sotto:
— MIRAGGI nella stanza 21: NORD e SUD erano scambiati.
— Non si doveva spaventare la BALENA;
— Non andare in acqua senza la BARCA;
— Non baciare la PUFFETTA.

TI Strange Castle
Prima di raccontarvi la solita storiella è doveroso informarvi di un particolare molto interessante sfuggitomi la volta scorsa. L'INPUT infatti dello scorso ADVENTURE, accetta ed accetterà tra verbo ed oggetto ogni sorta di articolo, aggettivo,

preposizione o altro...
Potete cioè impartire comandi del tipo:
— PRENDI LA MELA;
— PRENDI LA PICCOLA MELA;
— PRENDI QUELLA MELA;
— VAI SULLE MONTAGNE;
— VAI SULLE ALTE MONTAGNE
Etc... Comodo vero?
Ed ora torniamo alla nostra storia. Questa volta siete alle prese con un misterioro maniero che dovrete esplorare, per ritrovare dei gioielli rubati nascosti da qualche parte. Purtroppo non basterà ritrovare i preziosi ma dovrete cercare anche di uscire dalla miriade di corridoi che percorrono il castello in lungo ed in largo.

Uso del programma
Come per TI PUFFI anche questo adventure è composto da due programmi, la prima parte contiene tutto il vocabolario e le informazioni necessarie per iniziare l'avvetnura vera e propria che si trova sul secondo programma. Quindi:
1. Caricare il primo programma e dare il RUN.
2. Alla fine compariranno i soliti messaggi.
3. Non riavvolgere il nastro ma caricare di seguito tutta la seconda parte e, il gioco è fatto!!!
Vi do' appuntamento al prossimo numero ed ADVENTURE NOW!!!

■

GIOCHI DI SOCIETÀ
Si aprono nuove frontiere per i videogiochi multiutente con il recentissimo programma MUD che gira su DEC 10 della Essex University di Gran Bretagna. MUD (Multi Users Dungeon) è un'avventura che può essere vissuta contemporaneamente da più giocatori, seduti ognuno al proprio terminale, e che possono interagire nello svolgimento del gioco parlandosi, aiutandosi, o anche combattendo tra loro.
Lo scenario di questo "adventure game" è rappresentato da boschi, cittadelle, caverne e così via: tutti luoghi esplorabili per raccogliervi i tesori, sopravvivendo agli attacchi dei mostri in agguato.
Col passare del tempo maturerà anche il vostro livello d'esperienza: da novizio a guerriero, eroe, campione, superiore, incantatore, negromante, leggendario, fino a mago. L'essere mago comporta dei privilegi non indifferenti, come quello di poter essere immortali, controllare i mostri, strumentalizzare il gioco degli altri e - cosa più importante - avere sul proprio monitor una copia dello schermo di un qualsiasi altro giocatore, senza che questi ne sia al corrente; ancora: può imprigionare o escludere un avversario troppo rissoso, sapere - insomma - sempre e comunque tutto ciò che sta accadendo nell'avventura.
Il mainframe di MUD può essere contattato avendo un terminale munito di MODEM e software di comunicazione (bollette del telefono permettendo). MUD è un programma così complesso e imponente da dovere essere utilizzato solo dopo la mezzanotte o nei weekend, quando cioè l'attività dell'Università di Essex rallenta, per non lasciare a secco gli altri utenti.

È ARRIVATO DAL GIAPPONE IL QUICK DISK DRIVE
È la nuova concezione delle memorie di massa per i prossimi anni. Veloce e affidabile, il nuovo Quick Disk Drive prodotto dalla Yashica, si presenta con tutte le carte in regola per imporsi sul mercato dello home computer.
Da notare il formato ridotto del supporto magnetico, 2,8 pollici (7,1 cm), e il basso prezzo d'acquisto. La capacità del dischetto è di 128 Kbytes, il che rende il QD Yashica adatto a qualsiasi programma gestionale. Naturalmente è compatibile con qualsiasi apparecchio che porti il marchio MSX.

■

GLI ACCORDI ALLA CHITARRA

di ALFONSO BALDUCCI
e FRANCESCO CONTATORE

Nell'anno internazionale della Musica anche noi vogliamo dare un modesto contributo, dedicando questo programma a tutti i possessori di un TI-99/4A che vogliano "cambiare musica" e passare dalla tastiera del computer a quella più armonica della chitarra.
Una volta dato il RUN comporre la pagina di presentazione, quindi premendo un tasto si accederà alla schermata principale. A sinistra è riprodotto il manico della chitarra su cui il computer eseguirà gli accordi. Per esigenze diverse, sono visibili solo i primi 5 capitasto (le barrette poste sul manico) e quindi i primi 4 tasti (gli spazi compresi fra due capitasto). Quattro tasti sono sufficienti per visualizzare molti accordi, ma non tutti. È stato così necessario indicare accanto al primo capotasto visualizzato, il capotasto a cui l'accordo si riferisce:
(C I) significa che il primo capotasto visualizzato coincide col primo della chitarra;
(C IV) significa che il primo capotasto visualizzato si riferisce in realtà al quarto capotasto della chitarra, ecc.
Il computer, a questo punto, chiederà una scelta che potrà essere sia un accordo, nelle modalità e limiti che vedremo fra poco, sia una delle opzioni elencate più in basso: RIPETIZIONE, ACCORDO, ACCORDATURA CHITARRA, FINE.

Gli accordi

Gli accordi che il computer conosce sono la bellezza di 48 (quattro varianti per ciascuna nota, compresi i diesis). Infatti oltre gli accordi naturali (DO, DO♯, RE, RE♯, MI, FA, ecc.), il computer conosce anche gli accordi minori (DOm, DO♯m, REm, RE♯m, ecc.), gli accordi di settima (DO7, DO♯7, RE7, RE♯7, ecc.) e infine gli accordi minori di settima (DOm7, DO♯m7, REm7, RE♯m7, ecc.).
Nell'impostazione dell'accordo occorre rispettare l'ordine di priorità dei simboli ,m,7 (il computer giustamente non riconoscerà ad esempio REm♯, o SOL7m).
Una cosa molto importante è che l'impostazione dei simboli "♯" e "m" si ottiene premendo i tasti "♯" e "-" senza premere SHIFT (a tal proposito ho volutamente cancellato nel programma il simbolo "♯" vero e proprio per non incorrere in errori). Infine non bisogna porre spazi nel nome dell'accordo.
Molti spartiti, soprattutto di musica straniera, riportano l'indicazione degli accordi inglesi. Si tratta semplicemente di un diverso modo di chiamare le note e gli accordi secondo il seguente schema:
sistema italiano: LA SI DO RE MI FA SOL LA SI DO RE ecc.
sistema inglese: A B C D E F G A B C D ecc.
Il computer consentirà comunque anche l'impostazione di questi accordi.
Un'ultima cosa relativa agli accordi a proposito dei bemolle. Il computer non riconoscerà questi ultimi, tuttavia, trasformare gli accordi in bemolle in

accordi in diesis è cosa del tutto agevole seguendo lo schema di equivalenza sotto riportato:
accordi espressi in diesis: DO RE FA SOL LA
equivalenti in bemolle: REb MIb SOLb LAb SIb
Ovviamente MI diesis, SI diesis, FA bemolle e DO bemolle non esistono.

Esecuzione accordi
Impostato l'accordo sarà sufficiente premere ENTER affinché il computer lo esegua. L'esecuzione dell'accordo è sia visiva che sonora. Saranno visualizzati il posizionamento delle dita sulla tastiera della chitarra, la eventuale presenza del BARRÈ, il capotasto di riferimento e infine l'eventuale indicazione delle corde da non suonare, secondo la legenda posta in calce alla videata.
Per ripetere l'ultimo accordo impostato basta premere ENTER.

IL Barrè
Una nota particolare merita il BARRÈ. Il computer visualizza il barrè con due tondini bianchi (a volte anche più di due) su un solo tasto. Il barrè infatti consiste nel porre il dito indice lungo un tasto in modo da premere contemporaneamente più corde.
Il barrè realizza la trasposizione degli accordi. Ciò significa, ad esempio, che partendo dall'accordo di LA è possibile, spostando l'accordo di un tasto più in basso e ponendo il barrè sul primo tasto, ottenere l'accordo di LA , e ancora spostando il tutto di un altro tasto più basso ottenere l'accordo di SI ecc...
Il computer visualizza in maniera molto efficace la trasposizione degli accordi.
Esistono due tipi di barrè: il barrè completo, che si esegue ponendo l'indice in modo da premere tutte e sei le corde, e il DEMI-BARRÈ, che si esegue ponendo l'indice in modo da premere solo alcune corde. Nel primo caso si potranno suonare tutte le corde, nel secondo caso si dovrà omettere l'esecuzione delle corde che non risultino pigiate né dal barrè né da altre dita.
Il computer indicherà sempre il demi-barrè ma è chiaro che si potrà effettuare anche l'altro.

L'accordatura
Esistono diversi metodi di accordatura della chitarra. Premendo l'opzione 1 del programma il computer ne suggerisce contemporaneamente due.
La prima fase, che è comune ai due metodi, consiste nell'accordare la corda di LA (la seconda) avvalendosi di un altro strumento già accordato (flauto, organo, piano, nel nostro caso il computer). Il primo metodo consiste nel ripetere questa operazione anche per le altre corde. Perciò lo strumento dovrà fornire le note relative alle sei corde che sono:
MI basso, LA, RE, SOL, SI, MI cantino
Il secondo metodo consiste nel premere la corda di LA già accordata all'altezza del 5° tasto, ottenendo così un RE con cui si accorderà la corda successiva che è proprio quelle di RE. Quando si otterrà una uguaglianza di tono fra le due corde si dovrà pigiare la corda di RE all'altezza del 5° tasto in modo da ottenere un SOL con il quale si potrà la corda successiva che è proprio la corda di SOL e così via. Quando avremo accordato il MI cantino potremo accordare anche il MI basso effettuando il confronto fra le due corde fino a che i rispettivi suoni risulteranno simili.
Il computer eseguirà entrambe le procedure visualizzando il secondo metodo e fornendo le frequenze delle sei corde in modo da realizzare anche il primo metodo.
L'accordatura sarà indispensabile quando si voglia suonare insieme al computer.

■

ABBONATEVI A TI-99 NEWSOFT

Riceverete l'unica rivista in Italia per il TI-99/4A direttamente a casa vostra.

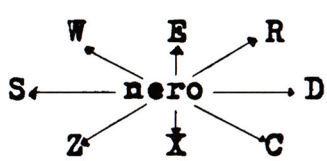
Per abbonarsi a 11 numeri basta inviare Lit. 95.000 con assegno non trasferibile o vaglia postale a:

NEWSOFT snc
Via S. Jacini, 4
20121 MILANO

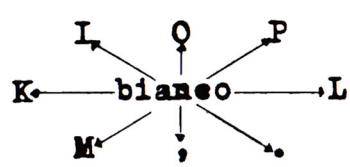
DUELLO

di SERGIO RINALDI

Questo gioco per 2 persone è ambientato nel vecchio west. Come premessa bisogna dire che per ottenere il massimo del divertimento da questo gioco sarebbe meglio usare i joysticks, comunque nel caso non li possediate, è previsto l'utilizzo della tastiera.
Dopo aver caricato il programma sullo schermo apparirà lo sfondo tipico del vecchio west con tanto di cow-boys armati di pistola e pronti a sfidarsi in una cruenta sparatoria.
Ogni giocatore guiderà il suo cow-boy usando la leva del joystick per dirigere il cammino nelle 8 direzioni consentite e il pulsante rosso per sparare contro l'avversario nella direzione corrispondente.
Ma attenti ai cactus e alla rocce presenti sul campo, i primi attutiscono il colpo scomparendo una volta colpiti, le seconde se colpite deviano il corso del proiettile ad effetto, quindi attenti a non incappare nel rimbalzo del vostro stesso proiettile.
Ogni avversario ucciso è un punto guadagnato; starà a voi quindi decidere di smettere premendo CTRL N o continuare CTRL S. Per quanto riguarda il comando del gioco da tastiera i tasti da usare sono i seguenti:



sparo Q



sparo Y

■

MYSTERY TOWER

di GIANLUIGI FEDELI

Mystery Tower è un gioco che attinge agli elementi tipici e agli intrecci del *Fantasy Game,* secondo una formula che consente al giocatore di confrontarsi con personaggi di miti, leggende e fiabe non solo attraverso una fascinazione di immagini che evocano gli scenari tipici del fantastico, ma anche con la proposta di enigmi che fanno da contrappunto all'azione del gioco.
Lo scopo di *Mistery Tower,* dunque, è classico e fedele all'ormai nota iconografia fiabesca: bisogna liberare la principessa rinchiusa da una strega malvagia in una torre maledetta; per fare questo ovviamente si richiede una certa attenzione, poiché nel corso del gioco vengono messi alla prova non solo i riflessi, ma anche il ragionamento.
Per liberare la principessa, il cavaliere dovrà superare gli ostacoli definiti sullo schermo secondo sette quadri che alternano all'azione propria degli ARCADE i suggerimenti e gli enigmi dell'ADVENTURE in una sintesi delle due formule che ho sviluppato secondo il TI-BASIC attraverso 518 istruzioni.
L'ultimo quadro è basato su un enigma la cui soluzione deve essere digitata secondo l'esempio:

BA = MENZOGNA;
C = VERITÀ

oppure

A = MENZOGNA;
CB = VERITÀ

A voi trovare la soluzione.

LISTATO:

10-550	Caratteri più stampa pagina grafica e istruzioni.
560-690	Entrata nella torre.
700-1620	Prima stanza.
1510-1570	Musichetta di vittoria.
1630-1750	Morte relativa al secondo quadro della prima stanza.
1760-1870	Routine Display at.
1880-2090	Presentazione seconda stanza.
2100-2280	Entrata seconda stanza + Definizione variabili + caratteri e colori + stampa sfondo.
2290-2370	Plottaggio rocce.
2380-2430	Movimento zattera.
2440-2600	Call Key + Movimento zattera
2610-2700	Morte nella seconda stanza
2710-2770	Vittoria e passaggio nella terza stanza.
2790-3020	Morte nel primo e nel terzo quadro della prima stanza.
3030-4070	Terza stanza.
4080-4720	Quarta stanza.
4720-4830	Enigma.
4840-5030	Vittoria e conclusione dell'avventura.
5040-5180	Musica iniziale e finale. ■

PLANETS HUNTER

di M.M.G. SOFTWARE

Planets Hunter è un tipico gioco spaziale multischermo ed è costituito da due programmi distinti, il primo contiene una schermata di presentazione del gioco e la definizione di alcuni caratteri grafici, il secondo contiene il gioco vero e proprio. Una volta caricato il secondo programma vi verrà richiesto se desiderate giocare con il joystick o con la tastiera, nel qual caso vi saranno richiesti i tasti che preferite usare per i vari movimenti.

Il gioco

Il gioco si compone di tre schermi diversi più uno schermo bonus. Nella prima fase di gioco dovete manovrare il vostro astronauta per aiutarlo a recuperare i pezzi del proprio razzo. Dopo aver raccolto un pezzo occorre spostarsi sulla destra dello schermo fino a che il pezzo raccolto non cade nella giusta posizione. I pezzi del razzo vanno raccolti in ordine: prima la base di appoggio, poi la parte bassa del razzo, quindi quella centrale, infine la cima.

Dopo aver completato il montaggio del razzo scenderanno dall'alto dei contenitori di carburante che dovrete raccogliere uno per volta e sganciare sul razzo passandovi sopra. Dopo aver completato il rifornimento di carburante il razzo partirà per portarvi nella fase successiva. Per completare la prima fase disponete di quattro schermi difensivi, dovete fare attenzione a non uscire dal teleschermo o a non scontrarvi con gli oggetti che lo attraversano orizzontalmente, pena la perdita di uno schermo. Potete anche distruggere le meteore e l'alieno che tentano di abbattervi sparando nel momento opportuno.

Nella seconda fase non correte il rischio di perdere l'astronave però il numero di volte che vi fate colpire influirà sul numero di astronavi di cui potrete disporre nella terza fase. In questa fase vi trovate in orbita intorno ad un pianeta che dovete liberare dagli alieni che lo infestano. Dovete evitare gli oggetti che si muovono verticalmente sullo schermo, potete anche sparare per distruggerli momentaneamente; attenzione però non tutti possono essere distrutti. Vi sono poi degli oggetti che attraversano lo schermo orizzontalmente, se li colpite otterrete dei punti ma lo scopo di questa fase del gioco è quello di portare la nostra astronave a posarsi su una delle lune che attraversano ogni tanto lo schermo. Quando sarete riusciti a scontrarvi con una luna scenderete e passerete alla terza fase.

A questo punto siete sulla luna che dovete liberare dagli alieni, dovete colire due volte ogni astronave nemica evitando i missili che vi lanciano e le astronavi suicide che cercano di speronarvi. Fate attenzione a non impiegare troppo tempo per distruggere tutte gli alieni altrimenti perderete un'astronave e dovrete ricominciare da capo.

Finito quest'ultimo schermo comparirà una grossa astronave che transitando in alto lascerà cadere dei piccoli para-

cadute che se raccolti vi daranno molti punti.
Quando anche questa fase si sarà conclusa dovrete nuovamente costruirvi un razzo per poter ripartire e andare a liberare altri pianeti.
Come potrete notare giocando il punteggio che ottenete non viene sempre visualizzato poiché in alcune fasi di gioco è molto importante la velocità ed è meglio non sprecare tempo per cose non indispensabili, comunque anche se non è visualizzato il punteggio viene ugualmente aggiornato e in caso di GAME OVER viene visualizzato il punteggio raggiunto fino a quel momento.
Passiamo ora ad esaminare il programma.
Il primo programma ha uno sviluppo lineare molto semplice che non è necessario analizzare in dettaglio, il suo compito principale è quello di definire alcuni caratteri grafici che non è stato possibile inserire nel programma principale.
Il secondo programma è quello che contiene il gioco vero e proprio.
Non è possibile fare un elenco delle variabili utilizzate poiché per risparmiare memoria esse sono state riutilizzate più volte all'interno del programma con compiti molto diversi tra loro.

100 - 120	contengono i dati relativi alle posizioni dei pezzi del razzo e dei nemici degli schermi successivi
130 - 170	qui vengono inizializzate alcune variabili dopo aver definito alcuni caratteri
80 -	se è la prima volta che viene eseguita permette di scegliere tra joystick e tastiera altrimenti passa oltre
190 - 220	scelta tra joystick e tastiere in quest'ultimo caso definizione dei tasti da usare
230 - 300	inizia la prima fase di gioco, viene disegnato lo scenario e posizionati gli sprite relativi alle parti del razzo, all'astronauta, agli asteroidi e all'alieno
310 - 340	con una IF si sceglie se leggere tastiera o joystick quindi viene messo lo sprite dell'astronauta
350 - 390	in queste linee vengono effettuati i controlli relativi alla costruzione del razzo e al suo rifornimento di carburante. Il NEXT della linea 390 completa il ciclo che costituisce la prima fase
400 - 630	fase di transizione allo schermo successivo con definizione di nuovi caratteri
400 - 630	fase di transizione allo schermo successivo con definizione di nuovi caratteri
640 -	inizia la seconda fase
650 - 660	viene letta la tastiera e mosso lo sprite dell'astronave
670 - 690	controllo degli oggetti che transitano orizzontalmente
700 - 830	passaggio alla terza fase, definizione di nuovi caratteri grafici e variabili
840 - 860	controllo movimento astronave
870 - 920	controllo astronave aliena e missile
930 - 940	routine per lo scroll laterale
950 - 1010	schermo bonus
1020 -	ricomincia da capo
1030 - 1040	routine GAME OVER
1050 - 1340	qui vi sono una serie di routine utilizzate dalle varie fasi di gioco per il controllo delle coincidenze e per lo sparo dell'astronave del giocatore
1350 -	routine a cui il programma salta il caso di errore; serve a evitare un'interruzione indesiderata del gioco.

■

STAR MATH

di RAFFAELE FERRIGNO

Questo programma si ispira al noto modulo *Number Magic* della Texas, di cui è certamente più povero ma forse più divertente, e vuole essere di ausilio a chi, avendo figli in età scolare e volendo rinfrescare le loro nozioni di aritmetica, ha il problema di rendere tale esercizio il meno noioso possibile. A tal fine contribuisce non solo la novità, peraltro ormai assai relativa, dell'uso del computer, ma anche un po' di quelle motivazione che continuano a decretare il successo di tanti giochi elettronici.

Venendo quindi al programma, che occupa circa 12,4 Kb e che quindi può creare problemi a chi usa l'unità a disco, dopo il titolo, che richiama in parte la dinamica del gioco, seguono una serie di scelte, cui si risponde, come d'altronde in tutto il gioco, con la semplice pressione dei relativi tasti numerici. In questo modo è possibile scegliere il grado di difficoltà delle operazioni proposte dal computer in relazione al livello degli studi del giocatore (i cinque livelli corrispondono, a norma dei programmi ministeriali, alle cinque classi della scuola elementare), il genere delle operazioni che si vuole eseguire (per il primo livello sono possibili solo addizione e sottrazione), il tipo di operazioni (del tipo A + B = ? oppure A + ? = B) ed infine la velocità del gioco da cui dipende il tempo che si ha a disposizione per le risposte.

Dopo di che il computer propone una serie di dieci operazioni. Ogni risposta errata viene rifiutata e, entro i limiti del tempo, si ha la possibilità di riprovare. A tempo scaduto invece lo score iniziale di 10 punti viene ridotto di un'unità e l'esatta risposta viene fornita dal computer. Lo score è sulla destra del quadro, mentre sulla sinistra viene mostrato il numero delle manches.

Come viene aggiornato lo score e come il computer conferma l'esattezza della risposta ve lo lascio scoprire nel corso della prova del programma.

Dopo ogni serie di dieci operazioni si ritorna alle scelte iniziali ed a questo punto è anche possibile interrompere il gioco scegliendo l'opzione 6 (END), che segue i livelli di difficoltà.

DESCRIZIONE DEL LISTATO

100-390	Titolo
400-490	Scelta del livello di difficoltà o della conclusione del gioco.
500-620	Scelta del genere di operazione (addizione - sottrazione - moltiplicazione - divisione)
630-680	Scelta del tipo di operazione (diretta - indiretta)
690-710	Scelta della velocità del gioco
720-790	Disegno del quadro
800-890	Scelta casuale dei numeri e disegno dell'operazione
900-930	Routine di accettazione della risposta
940	Routine di risposta errata
950-970	Routine di risposta esatta
980-1140	Routine di mancata risposta
1150-1170	Controllo del numero delle manches ed eventuale fine partita
1180-1260	Subroutines
1270	Subprogramma dei cicli di ritardo

Spettabile redazione,
mi chiamo Paolo e ho 15 anni. Vi scrivo per raccontarvi un fatto che mi capita spesso col Texas TI-99/4A: all'inizio il mio computer aveva un comportamento normale, ma quando comprai l'extended basic, dopo alcuni mesi notai strani blocchi del sistema operativo; per farlo diventare "normale", dovevo spegnerlo (neanche QUIT funzionava). Questo comportava la perdita del programma in memoria, che a volte era a molto lungo;
Vi chiedo se è possibile eliminare questo inconveniente; se sì, come?
Nell'attesa di un'eventuale risposta, colgo l'occasione per congratularmi per l'ottima rivista.
Cordiali saluti.
CAMPANELLI PAOLO
Bologna

* * *

...Approfitto della circostanza per chiedervi un chiarimento circa il modulo extended basic. Succede che spesso ed all'improvviso, nella digitazione dei programmi, il cursore si arresti senza alcuna motivazione con il risultato di perdere tutto il listato eseguito, con il disappunto che vi lascio immaginare. Nessun inconveniente si verifica invece nell'esecuzione dei programmi.
C'è una spiegazione a questo difetto ed è possibile ovviarci?
PIER VITTORIO GERBAUDO
Madonna dell'Olmo

Le strane anomalie denunciate dal computer derivano dal fatto che, come tutti gli apparecchi elettronici, anche il TI 99 è soggetto al fastidioso fenomeno dell'elettricità statica. Le cariche elettrostatiche presenti ovunque nell'aria, sul pavimento o su noi stessi provocano sbalzi di tensione a cui i delicati componenti del calcolatore non resistono e hanno per effetto quegli sconvenienti "blocchi" citati nelle lettere. Per minimizzare tali fastidiosi fenomeni bisognerebbe usare opportuni accorgimenti (non sempre efficaci) come evitare il più possibile di toccare le parti metalliche della consolle, non tenere su uno stesso piano metallico video, registratore e computer, assicurarsi del perfetto aggancio del modulo SSS (specie se è un modulo di espansione del linguaggio di programmazione come per l'extended) e così via. In questo modo col tempo non dovrebbero più sorgere problemi di questo genere.

M.S.

* * *

A TI 99 Newsoft
Sono un radioamatore ed uso il TI 99 per questa attività. Purtroppo però posso utilizzarlo solo come archivio dei dati che ci occorrono (collegamenti radio effettuati, calcolo delle rotte dei satelliti, ecc.), mentre non ho mai visto a portata di mano l'utilizzazione del nostro TI 99 come RTTY (radiotelescrivente) e per fare ciò ho dovuto acquistare un C64 il quale è fornito da tempo di un potente programma per la ricezione e trasmissione di segnali in telescrivente sia per i radioamatori sia per le varie Agenzie Stampa che ogni minuto irradiano le notizie via etere.
Inoltre non sono riuscito a trovare quali sono i terminali dei comandi "Clear to send", "Request to send", "Dati trasmessi" e "Dati ricevuti" nella parte del computer.
Come me ci sono molti radioamatori che usano il TI 99 solo limitativamente alla gestione di archivio e affidano ad altri la ricezione e la trasmissione di RTTY o Morse o altra forma.
Perché non chiedete ai lettori radioamatori di scrivervi per vedere se vale la pena studiare un programma per RTTY?
Oltre a ciò gradirei vedere sulla rivista schemi elettrici e di montaggio di tutto ciò che è accessorio per il TI 99: espansione, interfaccia, ecc.
Sono sicuro che se intraprenderete anche questa strada la nostra rivista ne trarrà solo vantaggi e anche l'entusiasmo per il TI 99 non calerà.
MARCO ELEUTERI
Roma

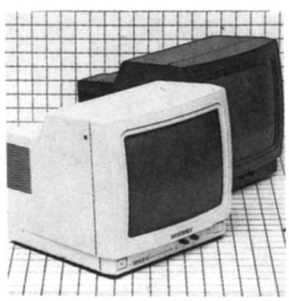
Accogliamo volentieri questo consiglio del Signor Eleuteri e ci appelliamo a tutti i radioamatori che sono in possesso del TI 99 e che seguono la nostra rivista, con la preghiera di scriverci a questo proposito. Anche per quanto riguarda gli schemi elettrici, molto presto potremo accontentare tutti coloro che ci hanno scritto o telefonato esprimendo il desiderio di vederli stampati sulla rivista.
Come sempre vi invitiamo a mandarci impressioni, suggerimenti e proposte per migliorare ancora di più la nostra/vostra rivista.

TI-99 COMPIÙMARKET

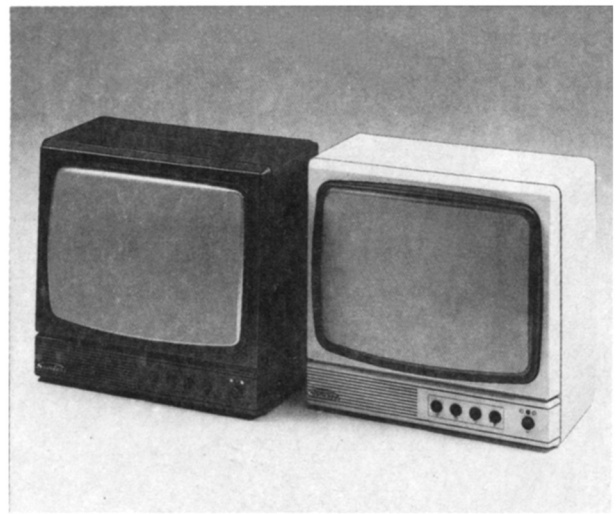
AC-04: Mobile per computer in bilaminato bianco; con rotelle e piano di lavoro rientrante. Studiato per alloggiare il vostro computer con tutte le periferiche, accessori, riviste, ecc. secondo una disposizione funzionale e versatile. Niente più fili volanti e spreco di spazio; ideale per la casa e per l'ufficio. Ideale anche il prezzo:
L. 320.000/290.000

AC-05: Power Master. Quadro di comando extra-piatto per computer e periferiche. Comandi separati per ogni singola apparecchiatura, interruttore generale, spia presenza rete, filtro antidisturbo. Posizionato sotto al monitor o alla stampante, si integra perfettamente con il vostro sistema.
L. 160.000/145.000

TV-01: Hantarex CT 900/1: monitor a colori 14'' con ingresso TTL (compatibile con il TI-99), risoluzione 320 × 290 pixel, audio incorporato, cavo di collegamento al TI-99 incluso.
L. 670.000/620.000



TV-02: Hantarex Boxer 12: Monitor monocromatico 12'' ad alta risoluzione. Audio incorporato, consumo minimo grazie all'alimentazione ''switched mode'', elevata stabilità di immagine, assenza di distorsioni geometriche, uniformità di fuoco su tutto lo schermo, mobile antiurto nei colori bianco o nero (specificare).
L. 280.000/250.000



TV-03: Sambers CTVM - 200/S 14: Monitor a colori con ingreso TTL (compatibile TI-99), risoluzione 360 × 576 pixel, audio incorporato, mobile in polistirolo antiurto nei colori bianco o antracite (specificare).
L. 610.000/560.000

TV-04: Sambers CTVMA 200/S 16: come sopra, 16''.
L. 630.000/580.000

TV-05: Sambers ''Mini'': TV color 16'' con ingresso monitor (presa scart). Telecomando ad infrarossi con funzioni Televideo e Stereo; 99 canali; 5 watt audio; antenna VHF-UHF incorporata; presa per antenna esterna 75 ohm; predisposto per funzione ''zoom'' (ingrandimento dell'immagine selezionabile dal telecomando); mobile in materiale plastico satinato nei colori grigio metalizzato o marrone metallizzato (specificare).
L. 820.000/750.000

TV-15: Modulo Zoom per TV color ''Mini''.
L. 35.000/30.000

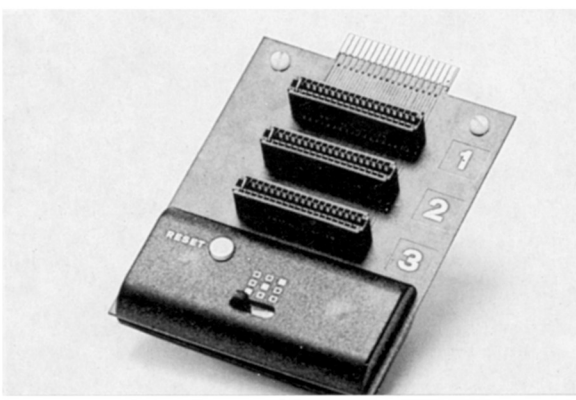
TI-99 COMPIUMARKET

MD-12: EXTENTENDED BASIC GRAFICO. Il nuovo Extended Basic II plus della Mechatronic offre, oltre alle prestazioni della versione precedente, due nuove serie di istruzioni:
Più di 20 nuove istruzioni, che non sono ottenibili con la versione standard e che semplificano sensibilmente lo sviluppo del programma e la velocità di esecuzione. Ad esempio routine di hardcopy, accesso diretto alla VDP, spostamento di blocchi di memoria, save e load di blocchi di memoria, ecc.
Grafica ad alta risoluzione. Permette di indirizzare ogni singolo pixel dello schermo usando tutti i 16 colori, sia nel disegno che nello sfondo. 40 potenti istruzioni grafiche permettono di tracciare linee, cerchi, rettangoli, archi, diagrammi circolari, ecc., di salvare su disco la grafica e di ottenere hard copy. Le istruzioni grafiche sono utilizzabili solamente con l'espansione di memoria 32K.
L'Extended Basic II plus viene fornito completo di manuale in italiano dell'extended standard, più manuale dettagliato delle nuove istruzioni.
L. 295.000/265.000

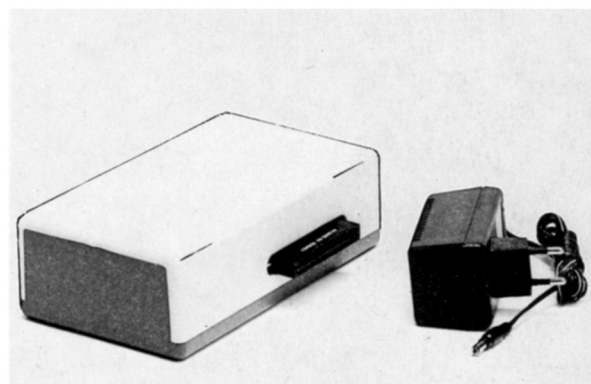
MD-22: Implementazione Extended Basic. Se già possedete l'extended basic Mechatronic, potete dotarlo del nuovo set di istruzioni del modulo II plus spedendocelo insieme alla cartolina market; l'implementazione verrà effettuata inserendo all'interno del modulo le RAM fornite dalla Mechatronic. La modifica include il manuale con il nuovo set di istruzioni.
L. 100.000/90.000

Attenzione: questa modifica non è possibile sul modulo originale Texas Instruments.

SC-12: Programma su cassetta per grafica ad alta risoluzione. Se possedete l'extended basic originale Texas, potrete comunque disporre della grafica ad alta risoluzione del modello II plus, caricando questo programma, che viene fornito completo di manuale di istruzioni.
L. 65.000/55.000



HW-08: Trislot. Scheda di estensione per moduli SSS. Si possono inserire tre moduli, selezionabili tramite un commutatore.
L. 95.000/85.000



HW-06: Espansione di memoria 32 KBytes RAM con alimentazione autonoma; non gravando sull'alimentatore del computer, ne evita il surriscaldamento durante il funzionamento prolungato.
L'alimentatore è incluso.
L. 240.000/220.000

HW-07: Espansione di memoria 32 KBytes RAM più interfaccia parallela con uscita standard Centronics in un'unico package applicabile lateralmente al computer (consente il collegamento in cascata di altre periferiche). Alimentatore esterno e cavo di collegamento alla stampante inclusi nel prezzo.
L. 380.000/350.000

A0166

TI 99 NEWSOFT

L. 10.000

Numero 13 - Anno II° - ottobre-novembre 1985

TI EXTENDED
SEA WAR
SPACE TI-99

PROGRAMMAZIONE
ADVENTURE NOW

comPIUmarket:
LE NOVITÀ
DEL MESE

SOMMARIO

☐ **PROGRAMMAZIONE**
Adventure Now — pag. 3

☐ **TI BASIC**
Concorso Ippico — pag. 5

☐ **TI EXTENDED**
Sea War — pag. 6

☐ **TI EXTENDED**
Space TI-99 — pag. 6

☐ **comPIUmarket** — pag. 8

SOFTWARE & SOFTWARE (LATO 1) N. 13

TI BASIC:
☐ Space TI
☐ TI Puffi
☐ Concorso Ippico

SOFTWARE & SOFTWARE (LATO 2) N. 13

EXTENDED BASIC
☐ Naufragio
(2 programmi)
☐ Sea War

TI 99 NEWSOFT

ANNO II° - Numero TREDICI
OTTOBRE-NOVEMBRE 1985
MENSILE
Direttore responsabile:
ANTONIO C. LOSITO
Coordinatore Editoriale:
VINICIO CIVININI
Direttore Amministrativo:
ENRICO CALLERIO
Art Director:
BEPPE RE-FRASCHINI
Illustrazioni:
MARIA TERESA COCOZZA
Segreteria:
CRISTINA FRISON
Redazione:
FIORENZA AURIEMMA
Collaboratori:
SERGIO BORSANI - GIANPAOLO BOTTIN - DANIELE CATALFAMO - PAOLO CIVARDI - STEFANO DELLI PONTI - FRANCESCO MACALI - MAURO MATTIAZZI - EZIO MONTINI - LEVIO PEROTTI - RICCARDO ROSSI - MARCO SQUINTANI - ALBERTO STRAFILE - SEBASTIANO TOMASELLO
TI 99 NEWSOFT è una pubblicazione della **NEWSOFT snc.**
Via Stefano Jacini, 4 - 20121 MI
Tel. 02/807464 - Telex 324284
REGISTRAZIONE AL TRIBUNALE DI MILANO N. 323 - DEL 7.7.84

© COPYRIGHT NEWSOFT snc
TUTTI I DIRITTI RISERVATI
Manoscritti e foto originali, anche se non pubblicati, non si restituiscono ed è vietata la riproduzione anche parziale di testi, fotografie e listati.

Pubblicità:

Tel. 02/862534
Abbonamenti: l'abbonamento annuo a 11 numeri di **TI 99 NEWSOFT** costa L. 95.000 per l'Italia; L. 160.000 per l'Europa (spedizione aerea); L. 200.000 per tutti gli altri paesi. L'importo, per assegno non trasferibile o per vaglia postale, va inviato a **NEWSOFT snc,** Via S. Jacini, 4 - 20121 Milano
Stampa: LALITOTIPO srl - Milano
Concessionaria per la distrib.:
EUROSTAMPA Srl - C.so Vittorio Emanuele, 11 - 10123 Torino
Telefono 011/538166-7

L'editore non si assume responsabilità alcuna sul contenuto degli articoli e/o software inviati. Per gli articoli firmati e/o siglati da collaboratori esterni, la redazione si assume la responsabilità prevista dalle leggi sulla stampa.

Tutti possono collaborare a TI 99 NEWSOFT, sia con articoli che con programmi. Scriveteci o telefonateci in redazione per avere tutti i ragguagli.

ADVENTURE NOW

di Daniele Catalfamo

Prima di inziare la nostra consueta ''chiaccherata'', mi sembra doveroso porgere un caloroso saluto a tutti coloro che ci hanno seguito fino ad oggi.
Questa puntata, infatti, conclude la nostra prima esplorazione del mondo degli ADVENTURS e, proprio per questo sarà più sostanziosa del solito.

COSTRUIRE UN'AVVENTURA

L'ultima cosa che mi resta da dirvi, ma non certo la meno importante, riguarda il salvataggio e relativo ''ripescaggio'' della situazione, cosiddetta corrente.
Infatti é molto comodo, ma soprattutto utile (alla salute!!), dopo aver trascorso metà della notte a cercare di scoprire ''qualcosa'', SALVARE la situazione corrente e ricaricarla, tale e quale, il mattino dopo, a mente ormai fresca e riposata.
Tutto ciò non é poi troppo complicato da farsi.
Noi, se ricordate, avevamo adottato la convenzione di assegnare ad ogni oggetto un numero:
— 1 se era in possesso del giocatore
— 0 se doveva essere ancora trovato, altrimenti il numero della stanza in cui si trova.
Inoltre ogni situazione speciale aveva anch'essa un numero associato:
Torcia accesa = 1; spenta = 0; porta aperta = 3; chiusa = 2.
ad esempio.
Poiché il giocatore opera solo su ciò che lo circonda, cioé le cose, basterà ricordare la situazione degli oggetti (quindi tutti i loro numeri) e le condizioni particolari.
Una volta impartito il comando opportuno, (SAVE di solito), il programma aprirà un file da registrare su cassetta, o per i piú fortunati su disco; il contenuto del FILE é il seguente:
a) La stanza in cui ci troviamo, o meglio la variabile che contiene il numero della stanza.
b) La situazione degli oggetti, quindi tutti i numeri relativi a questi ultimi.
c) Tutte le situazioni particolari o speciali, che si sono verificate fino a quel dato istante.
Per il file che carica la situazione corrente il discorso è del tutto analogo: l'unica differenza è il tipo del file che dovrà essere letto, anziché scritto.
In fig. 1 trovate un piccolo esempio di ciò che abbiamo illustrato. Con questo ritengo di avervi fornito una buona ''infarinatura'' e dei validi suggerimenti per la stesura di uno di questi giochi.

```
K = 5 stanza
F = 0 situazione particolare
0 (1) = 5 l'oggetto 1 é nella stanza 5
0 (2) = 1 l'oggetto 2 é in possesso del giocatore
0 (3) = 0 l'oggetto 3 deve essere separato.

10000 REMFILE da caricare
10010 OPEN 1:''CS1'', INTERNAL, INPUT, FIXED 192
10020 INPUT 1, K,F, 0(1), 0(2), 0(3)
10030 CLOSE 1'

10100 REM FILE da registrare
10110 OPEN 1. ''CS1'', INTERNAL, OUTPUT, FIXED 192
10120 PRINT 1: K, F, 0(1), 0(2), 0 (=)
10130 CLOSE 1
```

FIG. 1

GLI ADVERTURES TEXAS

Ci sono, o meglio VI ERANO, 11 titoli disponibili (Fig.2).
La loro scarsa diffusione é dovuta in primo luogo alla lingua inglese, e poi al ''possesso'' del cosiddetto modulo continuo ADVENTURE. Infatti, é necessario inserire nell'apposito stato il modulo ADVENTURE, (il quale contiene solo un programma di gestione), e poi caricare da cassetta (o da disco) una delle avventure. Sulle cassette sono registrati i dati relativi ad ogni avventura, i quali sono gestiti dal programma contenuto sulla cartuccia.
Le avventure di cui stiamo parlando non sono grafiche, sfruttano la gestione dello schermo a 40 colonne, e sono di buona qualità, infatti la classe di Scott Adams non é acqua!!
Menzione a parte deve essere fatta per l'avventura dal titolo RETURN TO PIRATE'S ISLE.
Questo adventure, che risiede su una cartuccia propria, é il proseguimento di PIRATE'S ADVENTURE, ma ciò che lo caratterizza é, senza ombra di dubbio il fatto di essere dotato di una grafica tridimensionale veramente NOTEVOLE.
Vi sono numerosissime stamze, tutte raffigurate in maniera eccellente, fin nei più piccolo particolari.
E adesso passiamo ad esaminare l'avventura seppure in maniera sommaria.
Il livello di difficoltà è superiore a quello delle avventure precedenti, e l'intera avventura é caratterizzata da una logica ferrea.
A tutti coloro che sono nei guai suggerisco di immedesimarsi nel personaggio e ''spremersi''il cervello.
E con questo abbiamo proprio finito. Ed ora, come al solito, vi lascio alle avventure della cassetta, si perché questo mese sono ben 2!!!, e vi invitiamo anche a dare un'occhiata all'appendice, in cui troverete dei suggerimenti per l'avventura PIRATE'S ADVENTURE.

TI PUFFI

Questa avventura, come ricorderete, era stata proposta in versione EXTENDED BASIC. Poiché i possessori di questo modulo sono ancora in minoranza, rispetto a coloro che posseggono soltanto le consolle, ho pensato di riscrivere l'avventura in TI BASIC, (un lavoraccio!!!), e il risultato lo potrete constatare voi stessi.
L'avventura, infatti, rispecchia fedelmente l'originale in tutto, (vi rimando al numero su cui era stato pubblicato inizialmente per le relative istruzioni), tranne che in alcuni messaggi.
Questi ultimi sono stati accorciati: infatti il TI BASIC, al contrario dell'EXT. BASIC, non offre la possibilità di righe multistatements, che cosentono di risparmiare memoria, ed accorciando messaggi ho risparmiato tanta memoria, quanta ne bastava affiché il programma ''girasse'.

ANALISI DEL LISTATO

100 - 340 Inizializzazione
360 - 560 Controllo dell'input
570 - 1610 Esegue il comando impartito
1620 - 1670 Stampa messaggio
1740 - 1900 Stampa situazione corrente
1910 - 2010 controlla se il VERBO e l'OGGETTO dell'input si trovano nelle rispettive tabelle
2020 - 2140 Inventario
2150 - 2410 Dati del programma
2420 - 2421 Accettazione input
2520 - 2580 Game Over

PRINCIPALI VARIABILI USATE

CO$ (10) tabella verbi
0$ (12) tabella oggetti
0 (12) numeri degli oggetti
S$ (36) luoghi/stanze
SS$ (36) descrizione stanze
S (36,4) matrice delle direzioni
F () matrice situazioni particolari

FIGURA 2
GLI ADVENTURE TEXAS
1) PIRATE'S ADVENTURE
2) ADVENTURELAND
3) MISSION IMPOSSIBLE
4) VOODOO CASTLE
5) THE COUNT
6) STRANGE ODYSSEY
7) MISTERY FUN HOUSE
8) PYRAMID OF DOOM
9) GHOST TOWN
10) SAVAGE ISLAND IR II
11) GOLDEN VOYAGE

APPENDICE
PIRETE'S ADVENTURE
Domande:
1) Come si costruisce la nave?
2) Come si apre la porta chiusa?
3) Come ci si libera del coccodrillo?
4) Come si salpa?
5) Come si conduce il pirata sulla nave?
6) Come ci si libera del serpente?
7) Dove si trovano i due tesori?
8) Come prendere il pesce senza farlo scappare via?

Risposte:
1) E' necessario possedere le istruzioni (PLANS), l'ancora, le vele, il legname.
2) Ma con le chiavi, che di solito si ripongono sotto il...
3) Semplice, basta nutrirli
4) Bisogna possedere la mappa ed una ciurma
5) Dategli ciò che vuole; da bere!!!
6) Dovete riporre le vostre speranze nel pappagallo
7) Si trovano entrambi sull'isola a cui si arriva salpando. Mi raccomando leggete bene la mappa!!!
8) Portate con voi anche l'elemento prezioso per l'animale (H2O) - Con tutti questi suggerimenti é un gioco da bambini risolvere l'avventura, vero?!!!

NAUFRAGIO

Tutti gli accorgimenti e i suggerimenti descritti negli articoli per la costruzione di un adventure, sono stati sfruttati in questo programma.
Dopo aver dato il RUN, prima di cominciare vi sarà chiesto se vorrete giocare una partita nuova, oppure no, in quest'ultimo caso dovrete caricare il ''vecchio'' file, per ristabilire la situazione in cui vi trovate quando é stato interrotto il gioco. E' altrettanto chiaro che in qualunque momento potrete salvare lo stato del gioco, con il comando SAVE GAME.
Inoltre anziché memorizzare le tabelle dei verbi e degli oggetti in opportune matrici, come era stato fatto per le avventure precedenti. Ho preferito non adottare tale metodo, ma con opportuna subruntines eseguo le operazione ed i controlli direttamente nelle linee date.
Cosi facendo ho risparmiato moltissima memoria arricchendo il programma con svariati e divertenti messaggi. E non é tutto!!!
Questo programma é infatti dotato di un comando ''HELP'' che a seconda della situzione può aiutarci oppure no, in ogni caso, provarci non nuoce!
Come sempre, tutte le istruzione e il prologo sono contenuti nella presentazione. La difficoltà é superiore a quelle precedenti, ma con un pò di buona volontà, fantasia ed i consigli dell'HELP credo che tutti porteranno a termine l'avventura.

Nota Bene - Come avrete notato, su questo numero é del tutto assente la soluzione di STRANGE CASTLE: non abbiate paura!!! La troverete prossimamente, sempre su TI 99 Newsoft. Ciò vale anche per NAUFRAGIO.

CONCORSO IPPICO

di Paolo Civardi

Il gioco che vi proponiamo stavolta simula un concorso ippico.
Sapete, quelle gare in cui impassibili cavalieri saltano ostacoli complessi, staccionate, muretti e piccoli fossati entro un'arena chiusa.
Pare difficile nella realtà e vi assicuriamo che lo é abbastenza anche nel gioco al compiuter.
Qui in redazione sono successe cose terribili durante le prove del gioco: i nostri cuori volavano regolarmente oltre l'ostacolo, il guaio era che il cavallo da noi guidato vi inciampa quasi sempre, i punteggi finali erano penosi.
Ma voi sarete degni della tradizione dei cavalieri italiani, dei D'Inzeo e Mancinelli e non avrete troppa difficoltà a diventare dei provetti campioni.

Regole e modalità:
Dovrete saltare gli ostacoli seguendo la numerazione delle bandierine poste accanto ad ogni staccionata.
Per ogni paletto abbattuto sarete penalizzati di quattro punti, viene definito percorso netto un percorso senza errori.
E' concesso girare liberamente per la pista senza eseguire alcun salto allo scopo di familiarizzarsi con il percorso.
Saltando troppo presto si cade da cavallo e si è squalificati cosi come ha un'alta probabilità di caduta cercando di saltare vicino ai sostegni delle barriere.

I comandi sono:
S per far girare il cavallo alla sua sinistra
D per far girare il cavallo alla sua destra
J per eseguire il salto
K per iniziare la corsa.
Bisogna poi porre attenzione al fatto che i comandi sinistra e destra si riferiscono alla prospettiva del cavallo e non alla vostra ma, anche se alle prime volte potreste confondervi, vedrete che poi questa convenzione vi sarà d'aiuto.

GLI ARRETRATI DI TI-99 NEWSOFT

Se avete perso qualche numero, o se avete scoperto da poco la nostra rivista, approfittate delle eccezzionali condizioni che TI-99 Newsoft vi offre per recuperare i numeri che vi mancano.

1 Numero arretrato L. 15.000

4 Numeri arretrati L. 40.000

10 Numeri arretrati L. 80.000

Per le richieste utilizzate l'apposita cartolina inserita nella rivista.

SEA WAR

di Gianpaolo Bottin

Questa volta invece del solito gioco spaziale vi presentiamo il rifacimento in EXTENDED BASIC di un gioco molto diffuso per il C 64 con il nome di BEACH HEAD. Naturalmente la versione per il TI 99 risente dei limiti imposti dalla lentezza del basic, rimane tuttavia un gioco godibilissimo anche se non eccezionale come velocità. Tra l'altro anche la memoria é sfruttata al massimo soprattutto per il fatto che abbiamo dovuto ridefinire per ben tre volte l'intero set di caratteri ad esclusione dei numeri e di gran parte delle lettere maiuscole.

Il gioco si compone di tre fasi distinte:

Nella prima fase dovrete distruggere quanti più aerei nemici potete. Per muovere il cannone antiaereo potete o usare il joystick oppure, se avete optato per la tastiera, i tasti:

E per alzare il tiro

S per spostare il cannone a sinistra

D per spostare il cannone a destra

X per abbassare il tiro

Q per sparare.

Ogni aereo che sfuggirà ai vostri proiettili bombarderà la flotta a vostra disposizione e dopo sei colpi inferti dal nemico perderete una delle navi. Viceversa per ogni aereo nemico distrutto guadagnerete 25 punti.

Dopo aver distrutto la flotta aerea del nemico dovrete affrontare i suoi cannoni, e si passa così al secondo schermo in cui manovrando il solito cannone dovrete cercare di distruggere le navi avversarie prima che esse abbiano il tempo di centrarvi con le loro cannonate. In questa fase sullo schermo é visualizzato un dato in più cioé l'alzo del vostro cannone che va da 0 a 45 gradi, a seconda dell'alzo potrete sparare più vicino o più lontano. Se il colpo che avete sparato va a vuoto vedrete sollevarsi degli spruzzi di acqua che vi permetteranno di migliorare la mira. Se sarete così bravi da distruggere tutte le navi nemiche potrete far sbarcare le vostre truppe per affrontare il terzo schermo. Avete a disposizione due carri armati per ogni nave superstite con i quali dovrete distruggere l'esercito nemico. Questo quadro é a scorrimento orizzontale, voi vi trovate sulla sinistra in basso e manovrando con il joystick o con i tasti **E** per saltare, **D** per accelerare, **S** per rallentare e **Q** per sparare dovrete evitare le buche e distruggere mine, camion ecc. il tutto facendo bene attenzione a evitare i proiettili che un elicottero vi sgancia dall'alto.

In questa fase guadagnate 100 punti per ogni nemico distrutto e dopo aver distrutto l'esercito avversario ricomincerete da capo la conquista di un'altra isola.

SPACE TI-99

di Paolo Tamburelli

Il gioco consiste nel cercare e distruggere le navi nemiche in una galassia divisa in 64 costellazioni diverse ogni partita, disposte come in figura 1.

Le navi avversarie sono disposte a caso nelle varie costellazioni e possono spostarsi durante il gioco.

Comandi a disposizione

A vostra disposizione avete una plancia di comando degna della più moderna delle astronavi. Alla richiesta del calcolatore di bordo ''COMANDO'' potrete inserire:

''LRS'' (sensori a lungo raggio) danno una visione di ciò che è contenuto nei quadranti intorno alla nave, adottando la simbologia indicata qui sotto.

Es.: 005

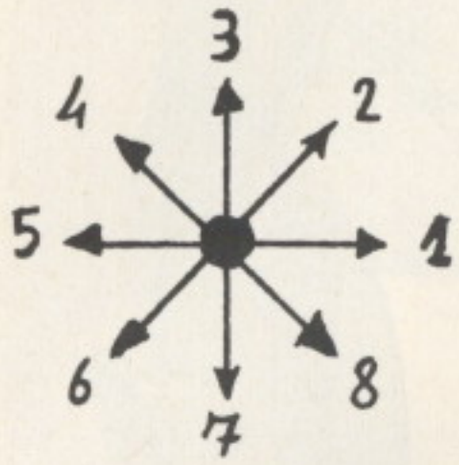
Il primo numero ci dice quante navi nemiche ci sono in quel quadrante.

Il secondo il numero delle nostre basi.

Il terzo il numero delle stelle.

''SRS'' (sensori delle stelle) disegnano sullo schermo la vostra posizione e aggiornano la data, il livello dell'energia, il numero di navi nemiche rimaste ecc.

''NAV'' (navigazione) appena inserito questo comando il computer di bordo vi chiederà l'angolo di rotta (compreso tra 1 e 8) la cui direzione corrispondente viene spiegata nel disegno.



Il calcolatore accetta anche valori non interi 1.56
N.B. - 1.56 e non 1,56
Una volta inserito l'angolo di rotta vi verrà chiesto il fattore iperluce (compreso tra 0 e 8 con la nave in piena efficienza, e tra 0 e 0,8 se la nave è risultata danneggiata in combattimento).
Per fattori inferiori ad 1 (esempio 2) si avranno spostamenti di un numero di caselle corrispondenti (esempio 2 caselle); per fattori superiori ad uno si effettuerà un balzo di un corrispondente numero di quadranti (esempio fattore 2 due quadranti).
Il calcolatore di bordo interromperà automaticamente la navigazione se lungo la rotta di accelerazione finirete contro una stella, una nave nemica o una base. Durante la navigazione è possibile incontrare delle deformazioni spazio temporali che vi faranno rimaterializzare in un punto diverso da quello calcolato. Il calcolatore di bordo vi impedisce pure di uscire dai limiti della galassia e di perdervi nello spazio.
''SHE'' (schermi protettivi) appena inserito questo comando il calcolatore vi chiederà la quantità di energia che vorrete destinare agli schermi, questa può essere recuperata se non viene usata in combattimento. In condizioni di allarme rosso il calcolatore vi avvisa se gli schermi non sono alzati. Se la nave viene colpita con gli schermi abbassati, viene inevitabilmente distrutta; se l'energia degli schermi è di poco superiore al colpo del nemico la nave può rimanere danneggiata in alcune parti (sensori, motori, ecc...), le riparazioni potranno durare anche diversi giorni astrali. Il calcolatore vi terrà informati.
''PHA'' (batterie di cannoni laser) sono batterie a puntamento automatico, appena inserito il comando il calcolatore vi chiederà con quanta energia volete fare fuoco. Una parte di questa (non tutta) colpirà il nemico e se sarà sufficiente ad abbattere i suoi schermi lo distruggerà, nel caso il colpo non fosse sufficientemente potente il calcolatore di bordo, tramite i suoi sensori saprà dirvi quanta energia è rimasta agli schermi della nave nemica.
''TOR'' (siluri) sono la componente più potente del vostro armamento, qualsiasi cosa colpiscano (comprese le vostre basi) la distruggono. Quando inserite questo comando il calcolatore vi chiede la rotta di lancio, che viene determinata come per la navigazione.
ATTENZIONE. Se inserite un comando diverso da quelli elencati il calcolatore lo considererà come NAVIGAZIONE.

CONSIGLI UTILI

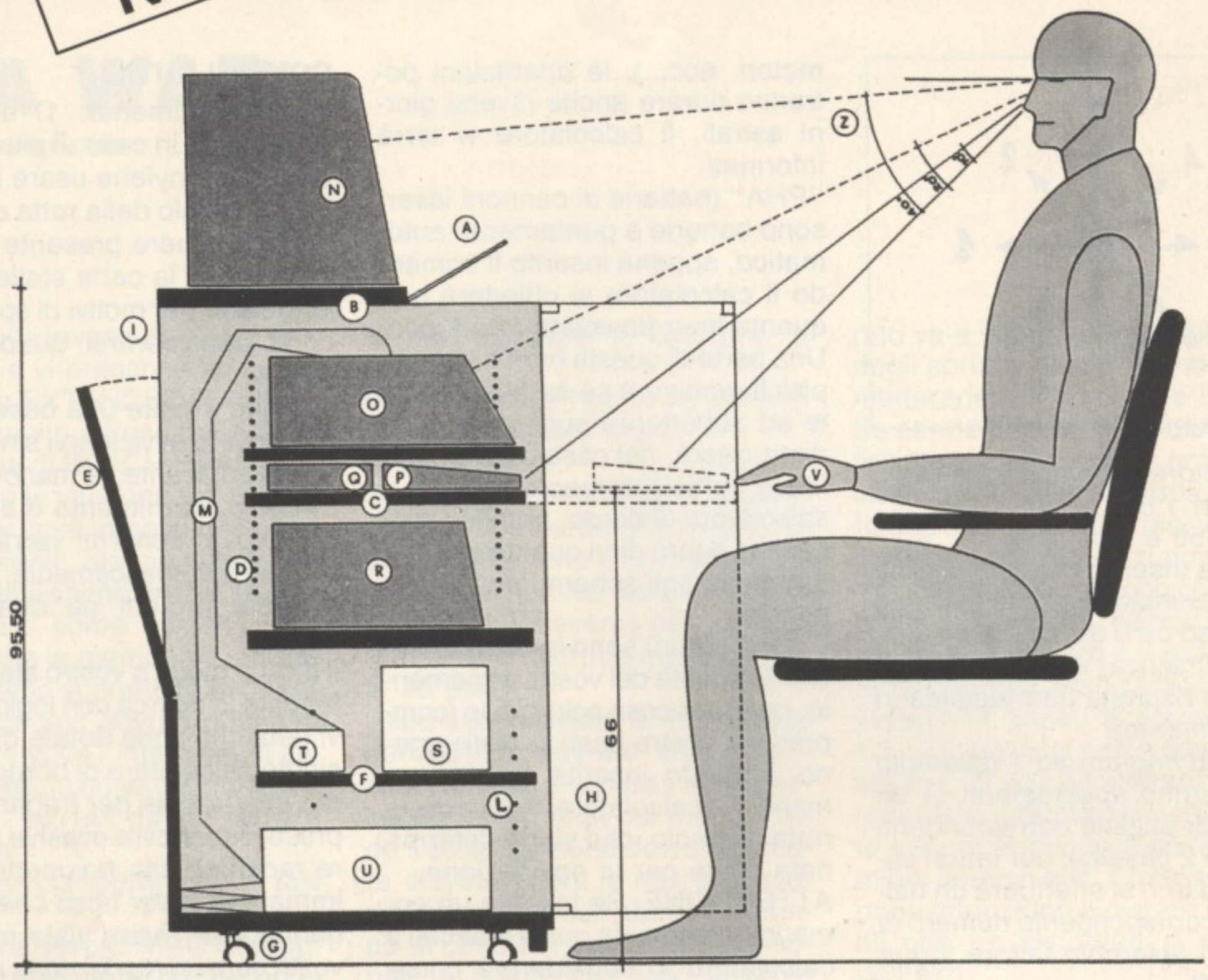
In combattimento: 1) alzare gli schermi, 2) in caso di più di un avversario conviene usare i phaser, 3) nel calcolo della rotta del siluro bisogna tenere presente che sullo schermo la carta stellare è rettangolare, per motivi di spazio, ma è da considerarsi quadrata nel calcolo.
Quando trovate una base appoggio, dovete avvicinarvi sino al quadretto adiacente, la manovra di attracco e rifornimento è automatica, i vostri schermi verranno abbassati automaticamente, in quanto siete protetti da quelli della base.
Il tempo gioca a vostro sfavore, effettuate la ricerca con logica, le navi nemiche sono dotate di un efficiente calcolatore di bordo che farà l'impossibile per fregarvi, comprese le manovre evasive e lasciare radioboe che trasmettono una immagine radar falsa che può ingannare la vostra vista ma non i vostri sensori (condizione rosso se la nave è vera, verde se è falsa).

Figura 1

INDI 1	INDI 2	INDI 3	INDI 4	VELA 1	VELA 2	VELA 3	VELA 4
REX 1	REX 2	REX 3	REX 4	APUS 1	APUS 2	APUS 3	APUS 4
ARA 1	ARA 2	ARA 3	ARA 4	VERA 1	VERA 2	VERA 3	VERA 4
DEA 1	DEA 2	DEA 3	DEA 4	CRUX 1	CRUX 2	CRUX 3	CRUX 4
LUPI 1	LUPI 2	LUPI 3	LUPI 4	LYRA 1	LYRA 2	LYRA 3	LYRA 4
CETI 1	CETI 2	CETI 3	CETI 4	GRUS 1	GRUS 2	GRUS 3	GRUS 4
LYNX 1	LYNX 2	LYNX 3	LYNX 4	PAVO 1	PAVO 2	PAVO 3	PAVO 4
LEO 1	LEO 2	LEO 3	LEO 4	DAX 1	DAX 2	DAX 3	DAX 4

NOVITÀ DEL MESE

TI-8[illegible]UMPIUMARKET



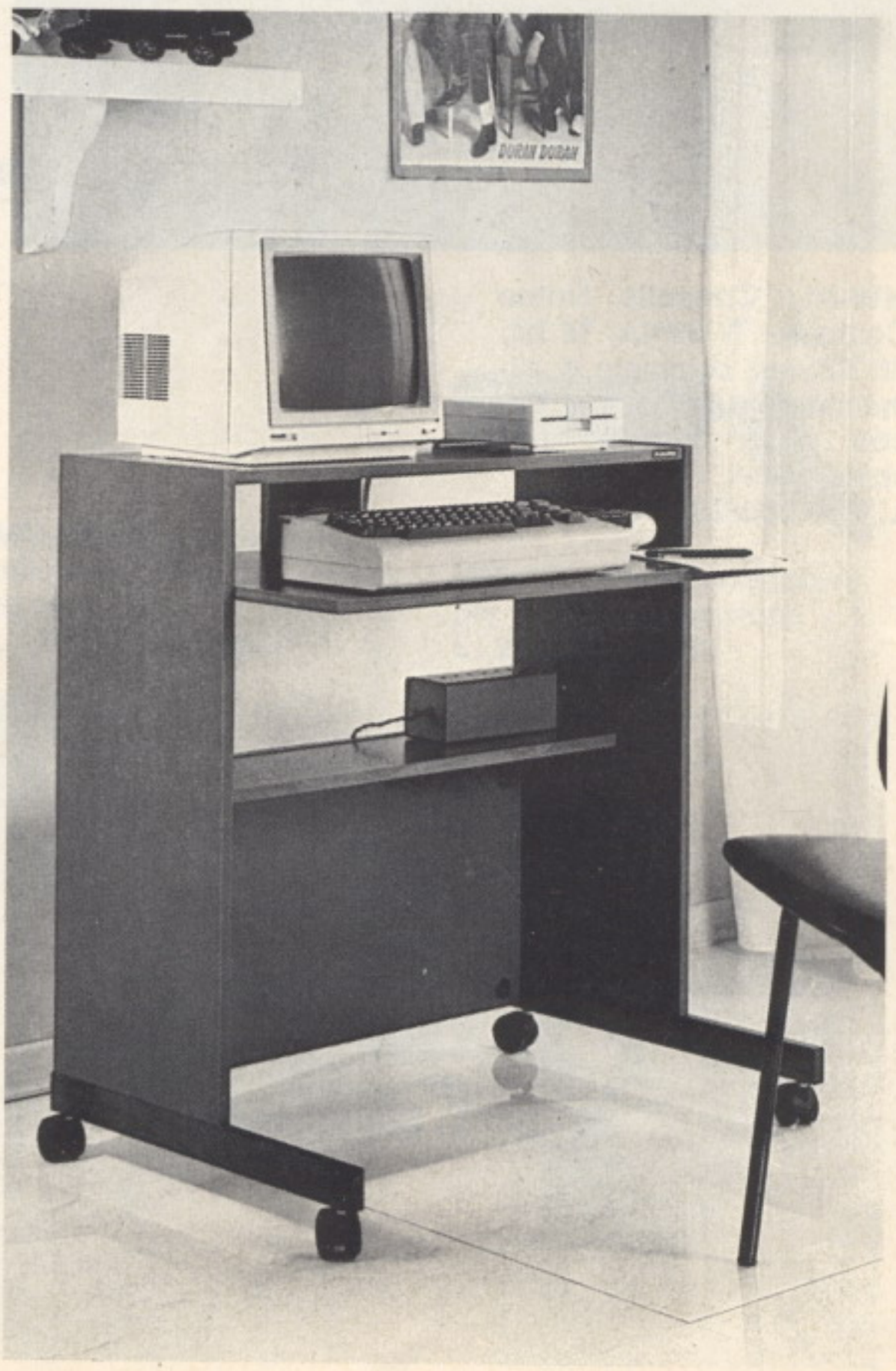
AC-08: Mobile portacomputer
A) Cristallo di chiusura superiore a molla frizionata, serve per caricare la carta sulla scrivente e per vedere lo scritto. B) Ripiano porta monitor e coperchio del mobile. C) Ripiano porta tastiera estraibile (profondità del piano cm. 40 uscita max. cm. 25, regolazione in altezza a gradini di cm. 16) scorrimento su cuscinetti senza grasso. D) Doppia foratura sia per la regolazione in altezza del piano porta scrivente sia per quello porta tastiera. E) Porta posteriore indispensabile per collegare gli apparecchi ed inserire la carta a modulo continuo. F) Piano regolabile per immagazzinamento carta a modulo continuo. G) Ruote piroettanti gemellate che consentono lo spostamento del mobile su tutti i pavimenti. H) Coppia porte anteriori in vetro con chiusura a calamita. I) Apertura superiore per la fuoriuscita aria calda e passaggio fili di collegamento. L) Foratura per la regolazione in altezza (mm. 32) del ripiano F. M) Percorso della carta. N) Posizione monitor su ripiano profondo cm. 30 e largo cm. 59. Q) Posizione per eventuali espansori. R) Posizione per floppy dischi o registratore. S) Spazio per monitor a riposo o altri apparecchi. T) Carta di alimentazione modulo continuo. U) Spazio per eventuale terminale o alimentatore. V) Altezza battitura a macchina regolabile a gradini di mm. 16. Z) Lettura verticale di tutti gli apparecchi, elimina completamente i movimenti laterale e verticale della testa e del collo. Colori: nero, noce, bianco (specificare nella richiesta).

L. 290.000/260.000

NOVITÀ DEL MESE

TI-99 COMPIUMARKET

AC-07: Mobile portacomputer.
Mobile con ruote, in bilaminato nero, noce o bianco (specificare nella richiesta). Misure h.94 × l.59 × p.55
L. 355.000/320.000



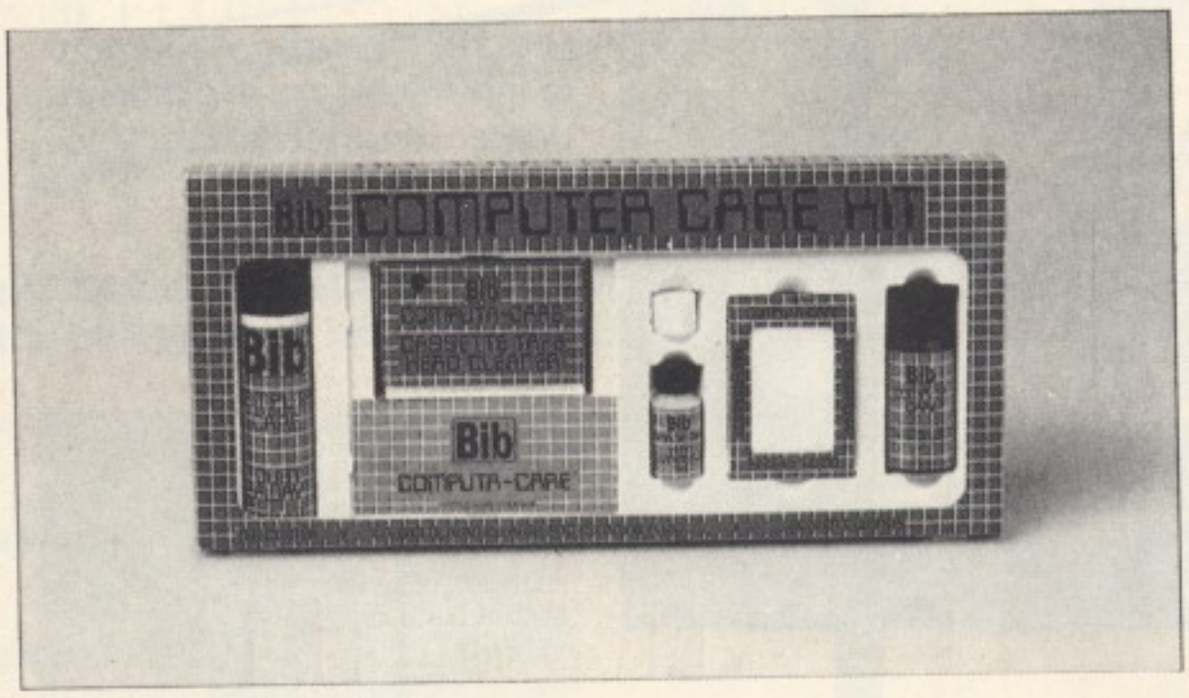
AC-06: Mobile portacomputer
Per configurazione senza stampante. Bilaminato nero, noce o bianco (specificare nella richiesta), con ruote. Misure h76 × l73 × p55.
L. 190.000/170.000

TI-99 COMPIUMARKET

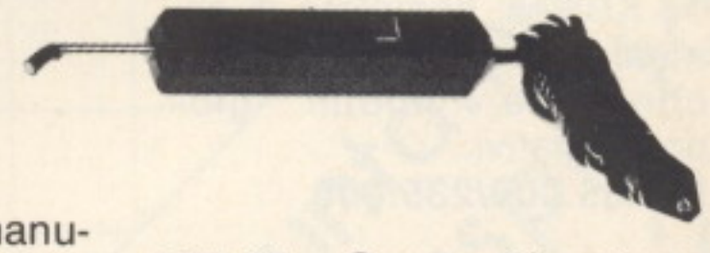
CS-02: Registratore a cassette Watson CR-5230, con VU-meter, alimentazione pile/rete.
L. 79.000/73.000

CS-11: Cavo di collegamento al registratore.
L. 12.500/11.500

CS-01: Registratore a cassette Irradio RE-730 Bit Recorder.
L. 72.000/69.000



MM-01: Kit pulizia e manutenzione computer e registratore a cassette.
L. 39.000/35.000



MM-02: Kit pulizia e manutenzione computer e disk driver.
L. 39.000/35.000

MM-03: Smagnetizzatore per testine registratore.
L. 29.000/26.000

HW-04: Joystick singolo (richiede cavetto adattatore).
L. 23.000/21.000

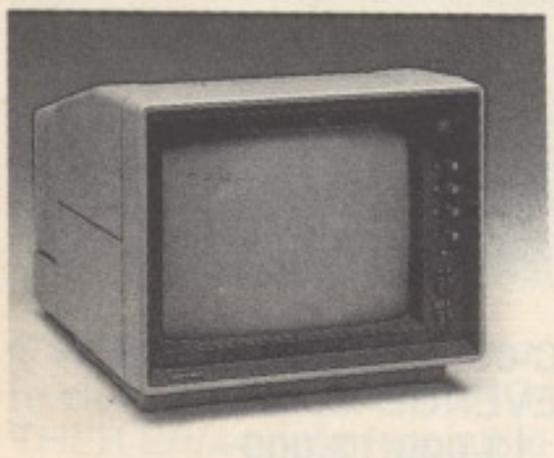
HW-14: Cavetto adattatore per due joystick.
L. 30.000/28.000

HW-05: Joystick professionale anatomico, con ventose, 4 pulsanti e autofire (richiede cavetto adattatore).
L. 28.000/31.000

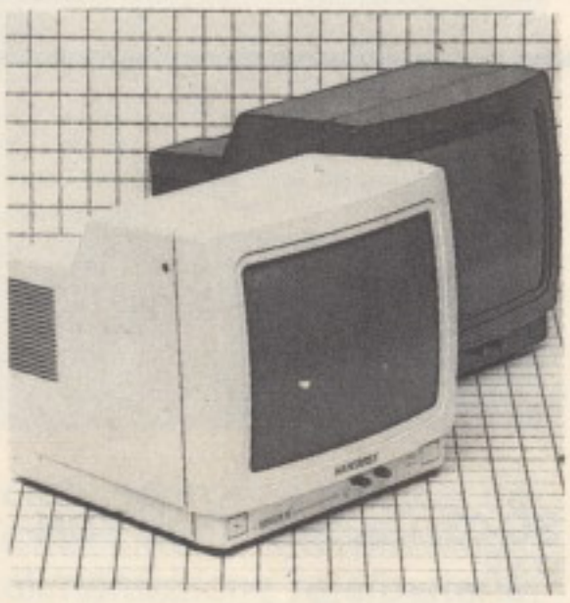
TI-99 COMPIUMARKET

AC-04: Mobile per computer in bilaminato bianco; con rotelle e piano di lavoro rientrante. Studiato per alloggiare il vostro computer con tutte le periferiche, accessori, riviste, ecc. secondo una disposizione funzionale e versatile. Niente più fili volanti e spreco di spazio; ideale per la casa e per l'ufficio. Ideale anche il prezzo:
L. 320.000/290.000

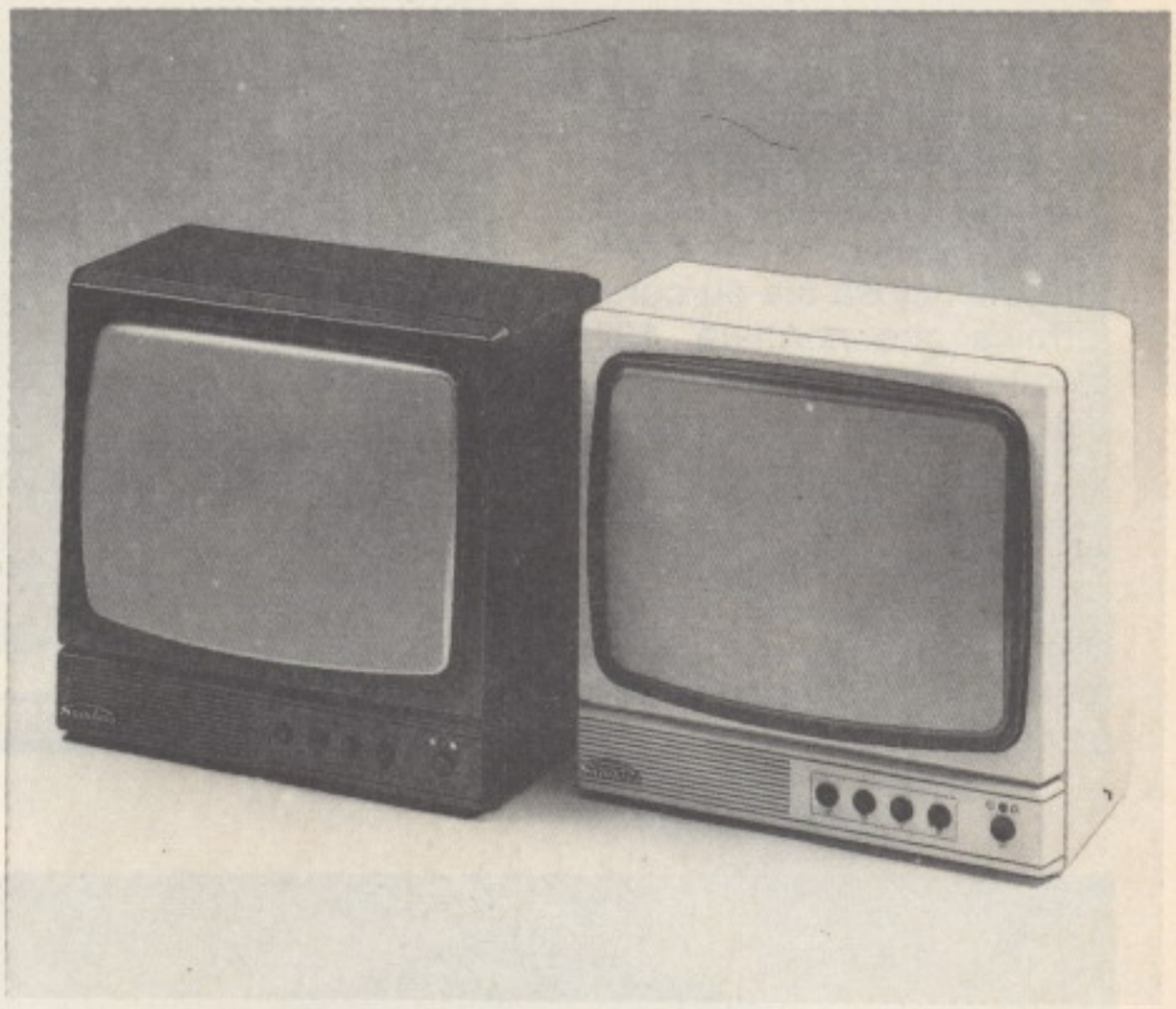
AC-05: Power Master. Quadro di comando extra-piatto per computer e periferiche. Comandi separati per ogni singola apparecchiatura, interruttore generale, spia presenza rete, filtro antidisturbo. Posizionato sotto al monitor o alla stampante, si integra perfettamente con il vostro sistema.
L. 160.000/145.000



TV-01: Hantarex CT 900/1: monitor a colori 14'' con ingresso TTL (compatibile con il TI-99), risoluzione 320 × 290 pixel, audio incorporato, cavo di collegamento al TI-99 incluso.
L. 670.000/620.000



TV-02: Hantarex Boxer 12: Monitor monocromatico 12'' ad alta risoluzione. Audio incorporato, consumo minimo grazie all'alimentazione ''switched mode'', elevata stabilità di immagine, assenza di distorsioni geometriche, uniformità di fuoco su tutto lo schermo, mobile antiurto nei colori bianco o nero (specificare).
L. 280.000/250.000



TV-03: Sambers CTVM - 200/S 14: Monitor a colori con ingreso TTL (compatibile TI-99), risoluzione 360 × 576 pixel, audio incorporato, mobile in polistirolo antiurto nei colori bianco o antracite (specificare).
L. 610.000/560.000

TV-04: Sambers CTVMA 200/S 16: come sopra, 16''.
L. 630.000/580.000

TV-05: Sambers ''Mini'': TV color 16'' con ingresso monitor (presa scart). Telecomando ad infrarossi con funzioni Televideo e Stereo; 99 canali; 5 watt audio; antenna VHF-UHF incorporata; presa per antenna esterna 75 ohm; predisposto per funzione ''zoom'' (ingrandimento dell'immagine selezionabile dal telecomando); mobile in materiale plastico satinato nei colori grigio metalizzato o marrone metallizato (specificare).
L. 820.000/750.000

TV-15: Modulo Zoom per TV color ''Mini''.
L. 35.000/30.000

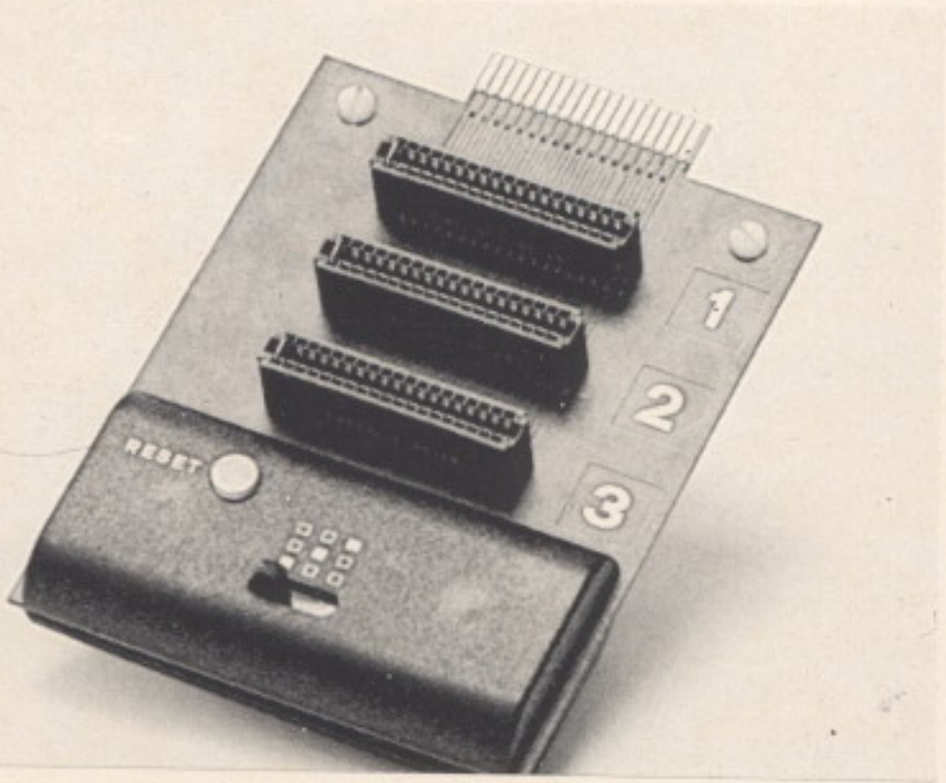
TI-99 COMPIUMARKET

MD-12: EXTENTENDED BASIC GRAFICO. Il nuovo Extended Basic II plus della Mechatronic offre, oltre alle prestazioni della versione precedente, due nuove serie di istruzioni:
Più di 20 nuove istruzioni, che non sono ottenibili con la versione standard e che semplificano sensibilmente lo sviluppo del programma e la velocità di esecuzione. Ad esempio routine di hardcopy, accesso diretto alla VDP, spostamento di blocchi di memoria, save e load di blocchi di memoria, ecc.
Grafica ad alta risoluzione. Permette di indirizzare ogni singolo pixel dello schermo usando tutti i 16 colori, sia nel disegno che nello sfondo. 40 potenti istruzioni grafiche permettono di tracciare linee, cerchi, rettangoli, archi, diagrammi circolari, ecc., di salvare su disco la grafica e di ottenere hard copy. Le istruzioni grafiche sono utilizzabili solamente con l'espansione di memoria 32K.
L'Extended Basic II plus viene fornito completo di manuale in italiano dell'extended standard, più manuale dettagliato delle nuove istruzioni.
L. 295.000/265.000

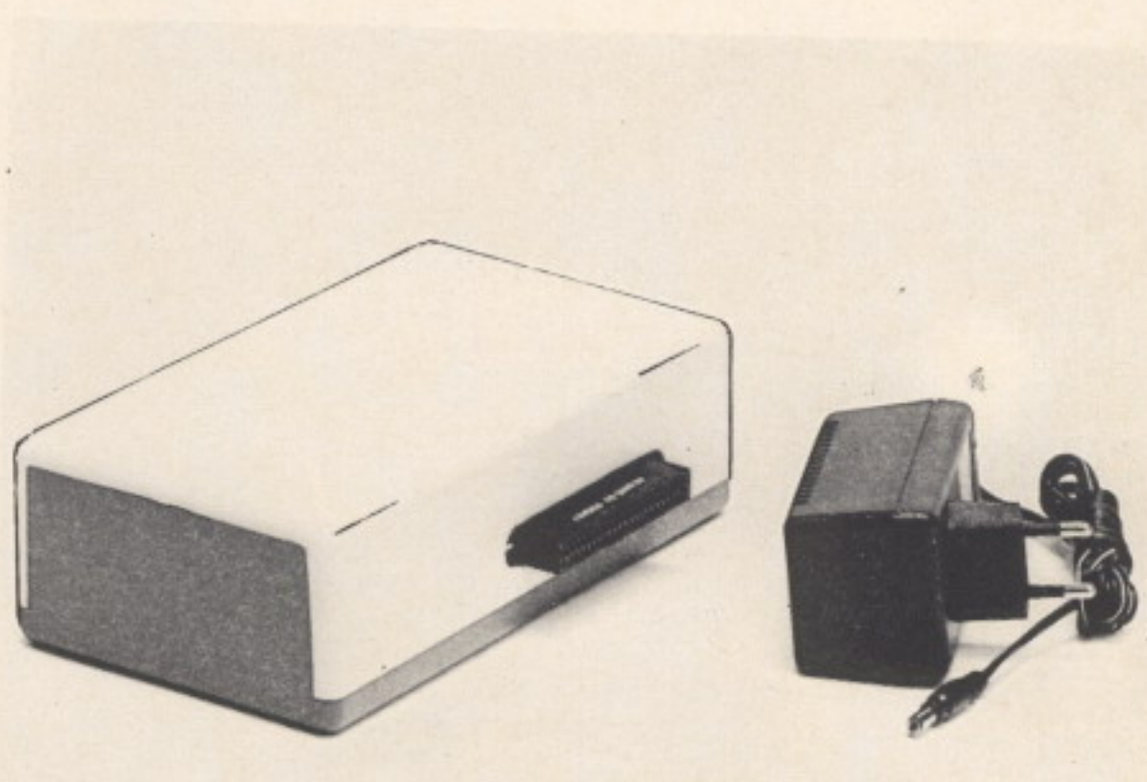
MD-22: Implementazione Extended Basic. Se già possedete l'extended basic Mechatronic, potete dotarlo del nuovo set di istruzioni del modulo II plus spedendocelo insieme alla cartolina market; l'implementazione verrà effettuata inserendo all'interno del modulo le RAM fornite dalla Mechatronic. La modifica include il manuale con il nuovo set di istruzioni.
L. 100.000/90.000

Attenzione: questa modifica non è possibile sul modulo originale Texas Instruments.

SC-12: Programma su cassetta per grafica ad alta risoluzione. Se possedete l'extended basic originale Texas, potrete comunque disporre della grafica ad alta risoluzione del modello II plus, caricando questo programma, che viene fornito completo di manuale di istruzioni.
L. 65.000/55.000



HW-08: Trislot. Scheda di estensione per moduli SSS. Si possono inserire tre moduli, selezionabili tramite un commutatore.
L. 95.000/85.000



HW-06: Espansione di memoria 32 KBytes RAM con alimentazione autonoma; non gravando sull'alimentatore del computer, ne evita il surriscaldamento durante il funzionamento prolungato.
L'alimentatore è incluso.
L. 240.000/220.000

HW-07: Espansione di memoria 32 KBytes RAM più interfaccia parallela con uscita standard Centronics in un'unico package applicabile lateralmente al computer (consente il collegamento in cascata di altre periferiche). Alimentatore esterno e cavo di collegamento alla stampante inclusi nel prezzo.
L. 380.000/350.000

TI 99 NEWSOFT

L. 10.000

Numero 14 - Anno II° - dicembre 1985

EXTENDED + 32K:
ISIGRAPHICS

HARDWARE:
IL TASTIERINO
NUMERICO

MINI MEMORY:
SNAKE-SNAKE

SOMMARIO

☐ **GENETICA**
La legge di Hardy-Weinberg — pag. 3

☐ **HARDWARE**
Il tastierino numerico — pag. 4

☐ **TI EXTENDED**
Grotte oscure
Abissi — pag. 6

☐ **TI EXTENDED + 32 K**
Easy Graphics — pag. 7

☐ **TI BASIC**
Circuiti Logici — pag. 8

☐ **MINIMEMORY**
Snake - Snake — pag. 10

☐ **RECENSIONI** — pag. 11

☐ **comPIUmarket** — pag. 12

SOFTWARE & SOFTWARE **(LATO 1)** **N. 14**

TI BASIC:
☐ Circuiti Logici
☐ Genetica (2 Programmi)

MINI MEMORY:
☐ Snake Snake (2 Programmi)

SOFTWARE & SOFTWARE **(LATO 2)** **N. 14**

EXTENDED BASIC
☐ Caverne Oscure (2 Programmi)
☐ Abissi

EXTENDED + 32 K
☐ Easy Graphics (3 Programmi)

TI 99 NEWSOFT

ANNO II° - Numero QUATTORDICI

MENSILE

Direttore responsabile:
ANTONIO C. LOSITO
Coordinatore Editoriale:
VINICIO CIVININI
Direttore Amministrativo:
ENRICO CALLERIO
Art Director:
BEPPE RE-FRASCHINI
Illustrazioni:
MARIA TERESA COCOZZA
Segreteria:
CRISTINA FRISON
Redazione:
FIORENZA AURIEMMA
Collaboratori:
SERGIO BORSANI - GIANPAOLO BOTTIN - DANIELE CATALFAMO - PAOLO CIVARDI - STEFANO DELLI PONTI - FRANCESCO MACALI - MAURO MATTIAZZI - EZIO MONTINI - LEVIO PEROTTI - RICCARDO ROSSI - MARCO SQUINTANI - ALBERTO STRAFILE - SEBASTIANO TOMASELLO

TI 99 NEWSOFT è una pubblicazione della **NEWSOFT snc.**
Via Stefano Jacini, 4 - 20121 MI
Tel. 02/807464 - Telex 324284
REGISTRAZIONE AL TRIBUNALE DI MILANO N. 323 - DEL 7.7.84

© COPYRIGHT NEWSOFT snc
TUTTI I DIRITTI RISERVATI
Manoscritti e foto originali, anche se non pubblicati, non si restituiscono ed è vietata la riproduzione anche parziale di testi, fotografie e listati.

Abbonamenti: l'abbonamento annuo
a 11 numeri di **TI 99 NEWSOFT** costa L. 95.000 per l'Italia; L. 160.000 per l'Europa (spedizione aerea); L. 200.000 per tutti gli altri paesi. L'importo, per assegno non trasferibile o per vaglia postale, va inviato a **NEWSOFT snc,**
Via S. Jacini, 4 - 20121 Milano
Stampa: LALITOTIPO srl - Milano

L'editore non si assume responsabilità alcuna sul contenuto degli articoli e/o software inviati. Per gli articoli firmati e/o siglati da collaboratori esterni, la redazione si assume la responsabilità prevista dalle leggi sulla stampa.

Tutti possono collaborare a TI 99 NEWSOFT, sia con articoli che con programmi. Scriveteci o telefonateci in redazione per avere tutti i ragguagli.

LA LEGGE DI HARDY-WEINBERG

di CLAUDIO MACCONE

Probabilmente molti lettori avranno già sentito parlare, a scuola nei corsi di scienze naturali, delle leggi di Mendel che regolano il passaggio dei caratteri ereditari da una coppia di genitori ai loro figli. Bene: sul nostro TI 99 possiamo simulare queste leggi, rendendone così la comprensione molto più semplice. Basterà che voi seguiate passo per passo quanto spiegato in queste tre pagine. Dunque mettetevi a vostro agio, accendete il computer, e, dopo la scritta TI BASIC READY, battete PRINT RND. Come vedete è comparso un numero del tipo ''zero virgola qualcosa'', cioè, come si dice nella notazione anglo-americana, ''point something''. Questo è un numero casuale (= random number) ed è sempre maggiore di zero e minore di uno. Potete ripetere l'operazione molte volte: di solito verrà fuori ogni volta un numero diverso, ma sempre compreso fra zero ed uno. Se invece volete produrre un numero casuale intero compreso fra due numeri prefissati, ad esempio fra 3 e 10, allora battete PRINT INT (10 - 3 + 1) * (RND + 3), espressione che diventa subito INT (B - A + 1) * (RND + A) se gli estremi sono i due numeri A e B (dove A < B).

Marcia a caso

Il primo programma che vi proponiamo è così breve che ne potete vedere il listato in una sola videata. Esso simula (cioè imita) il lancio di una moneta: può venire solo o testa o croce, e se viene testa il computer stampa + 1, mentre se viene croce stampa —1. Siete voi a decidere quante volte, cioè quanti lanci della moneta, volete fare. Finiti tutti i lanci, il computer calcola la somma algebrica dei +1 e dei —1, e la scrive: è chiaro intuitivamente che questo numero dovrebbe essere zero, ma non è detto che in pratica risulti proprio zero. Allora diciamo che zero è il valor medio teorico delle nostre simulazioni, mentre il numero ottenuto realmente è il valor medio empirico. Inoltre il lettore dovrebbe fare attenzione alle linee 170-180: il nostro numero casuale C non è intero ma reale (cioè con la virgola e i decimali) ed è compreso fra zero e due; se esso è minore di 1 verrà testa, altrimenti verrà croce: tutti i numeri fra zero e due hanno la stessa probabilità di venir prodotti, ossia la loro distribuzione è uniforme.

Prove di Bernoulli

Il secondo programma è un'estensione del precedente al caso in cui la probabilità che venga testa non è più uguale a quella che venga croce, ma può essere fissata a piacere da voi. La probabilità di un evento è un certo numero, compreso fra zero ed uno, che ci indica qual è la possibilità che l'evento si verifichi. Per esempio, voi dovrete digitare 25 se supponete che l'evento simulato abbia il 25% di probabilità (P) di avverarsi. Poi dovrete dire al computer quante prove (N) volete eseguire ed allora questo stamperà un ''si'' od un ''no'', a seconda che l'evento di probabilità 25 si verifichi o no in ciascuna prova, inoltre esso stamperà un 1 nel primo caso, ed uno zero nel secondo, facendo poi la somma ditutti gli uno, cioè del numero di volte in cui l'evento si è verificato. Il matematico svizzero Jakob Bernoulli (1654-1705) dimostrò che il valor medio teorico di questa somma è NP, mentre la deviazione standard teorica è V NP (1-P). Questi valori sono calcolati e stampati dal computer affinché vediate che il valor medio empirico, cioè la somma degli uno, è davvero prossimo al valor medio teorico, e che la differenza fra di essi è dell'ordine delle deviazioni standard. Anzi, l'accordo diventa sempre più buono se si aumenta il numero N delle prove (legge dei grandi numeri di Bernoulli).

Legge di Hardy-Weinberg

E veniamo alla genetica. Si dice gene un carattere che viene trasmesso dai genitori ai figli. Ad esempio, se i due genitori hanno entrambi gli occhi azzurri, è assai probabile che anche i loro figli abbiano gli occhi azzurri, e allora si chiamerà ''occhi azzurri'' il gene trasmesso. Abbiamo detto ''assai probabile'' e non ''certo''. Infatti ciascuno dei due genitori possiede non un gene soltanto, ma DUE, perché ha avuto due genitori. Se l'altro gene è ''occhi verdi'', ma è recessivo rispetto a quello ''occhi azzurri'' il genitore avrà gli occhi azzurri, ma porterà nel suo corpo anche il gene ''occhi verdi'', e lo potrà benissimo trasmettere ai suoi figli. I figli, però, avranno gli occhi verdi solo se ENTRAMBI i due geni sono del tipo ''occhi verdi''.

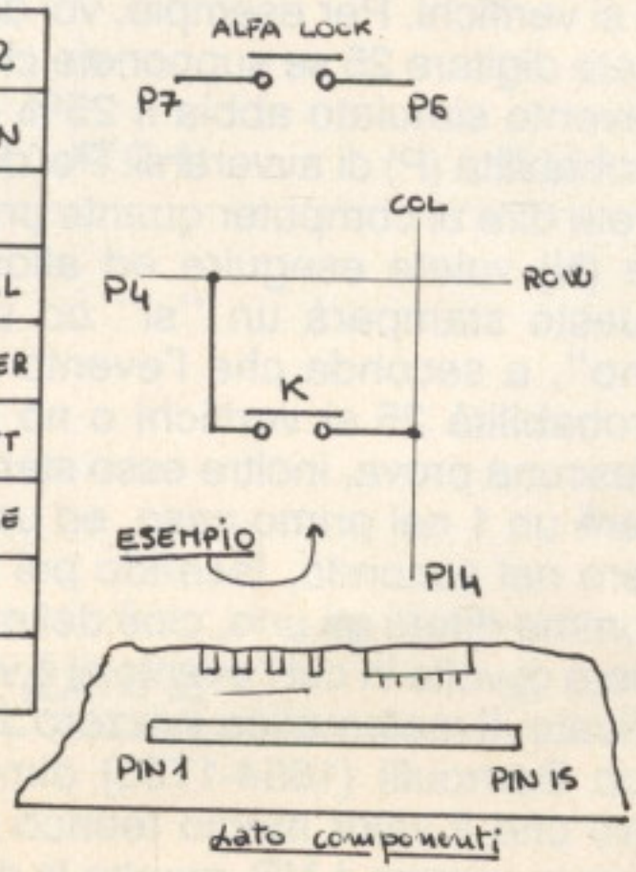
Il terzo programma che vi proponiamo simula tutto ciò sul nostro TI-99. Voi scegliete il numero (N) di individui, cioè di figli, che volete mettere al mondo. La legge di Hardy-Weinberg, scritta sopra, suppone, che il numero N non cambi fra una generazione e l'altra. Quindi anche i genitori saranno in tutto N (cioè N/2 maschi ed N/2 femmine). Siccome ogni genitore ha DUE Alleli, in tutto gli alleli saranno 2N. Voi scegliete anche quanti di questi 2N alleli, sono del tipo ''occhi azzurri'', cioè A (che è l'input alla linea 110, e deve essere minore di 2N). La probabilità p dell'allele A è dunque A/2N (linea 120), mentre quella dell'allele q (''occhi azzurri'') è 1-p (linea 130). Il computer accoppia a caso i due alleli per ogni individuo, e alla fine stampa i risultati.

IL TASTIERINO NUMERICO

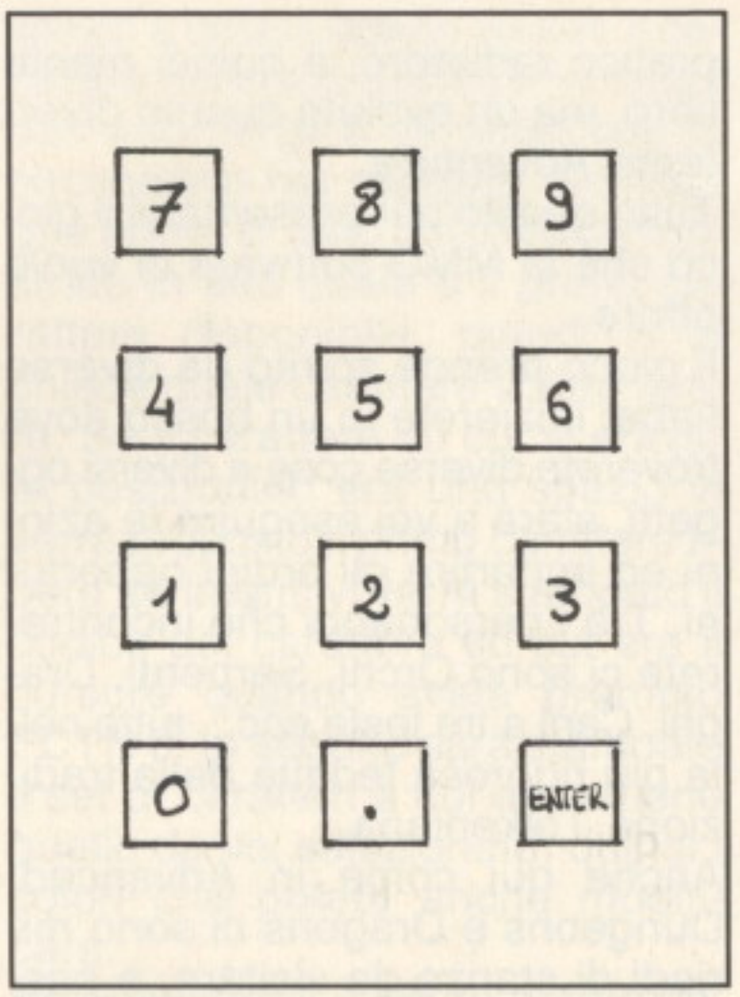
di ENRICO MATTEUCCI

Quello che vi proponiamo di realizzare in questo numero è uno strumento veramente utile. Sicuramente avrete già notato che molti personal e Home computer, oggi in commercio, sono dotati insieme alla normale tastiera QWERTY, anche di una serie di tasti che ripetono i numeri già presenti sulla fila superiore della tastiera stessa.
Questa duplicazione di tasti è appunto quel che si chiama un TASTIERINO NUMERICO; quindi perché non dotare anche il nostro TI 99 di questo utilissimo optional?
Vediamo brevemente quali sono i vantaggi che si ottengono quando si usa un tastierino numerico (TN):
— TN è particolarmente indicato nell'utilizzo di programmi professionali come TI-Calc oppure programmi di statistica.
— Quando sia comunque richiesta l'immissione di una grossa mole di dati numerici.
— I lunghissimi DATA (es. per le definizioni dei caratteri grafici) nei programmi basic.
— Gli interminabili CALL LOAD presenti nei programmi che usano l'espansione di memoria.
E infine, ma non per questo meno importante, la possibilità per i mancini di posizionare TN alla loro sinistra con gli ovvi vantaggi che ne possono ricavare.
La spesa per la realizzazione di TN è irrisoria, può andare dalle poche migliaia di lire ad un massimo di quindici-ventimila lire se i tasti che acquisterete saranno di tipo professionale.
Tutto quello che dovrete comprare sono infatti dodici tasti (interruttori normalmente aperti) ed un semplice contenitore su cui installare il tutto.
Si potrà anche riciclare del materiale surplus già in vostro possesso, ma vista l'esigua spesa è forse meglio agire con materiale sicuramente funzionante.
Per gli eventuali abbellimenti estetici tutto è lasciato alla fantasia del costruttore.
Passiamo ora ad una breve descrizione del funzionamento della tastiera del TI 99 che è collegata alla piastra madre del computer tramite la porta periferica J3 (KEY BOARD).
Questa porta è composta da 15 pin numerati a partire da sinistra (per maggiori dettagli vedi n. 10 di TI 99 New Soft giugno 1985, oppure figura 1).
È quindi possibile collegarci, in parallelo alla tastiera, a questa porta per ottenere una duplicazione dei tasti numerici.
Cortocircuitando ordinatamente coppie di pin della porta J3 si possono ottenere tutte le lettere, i numeri e le funzioni della tastiera secondo la tabella di figura 1.
TN avrà dodici tasti che opereranno questi collegamenti:
— 10 tasti per i numeri;
— 1 tasto per la duplicazione dell'ENTER;
— 1 tasto per la duplicazione del punto (.).
La figura 2 rappresenta la disposizione dei tasti in un TN in versione standard.
Passiamo ora ad una descrizione della realizzazione pratica del progetto.
Lo schema elettrico del TN è quello di figura 3, dove sono rappresentati tutti i collegamenti necessari per il suo corretto funzionamento.
Per la realizzazione fisica sarebbe utile anche un circuito stampato, ma visto che ciò comporterebbe altri problemi, pensiamo che se agirete con calma realizzando un collegamento alla volta (cancellandolo corrispondentemente sullo schema) non dovrebbero sorgere complicazioni.
Potete osservare sullo schema di figura 3 che ogni tasto (rappresentato con il simbolo X) chiude il collegamento di due pin sulla porta J3 secondo l'ordine di fig. 1.

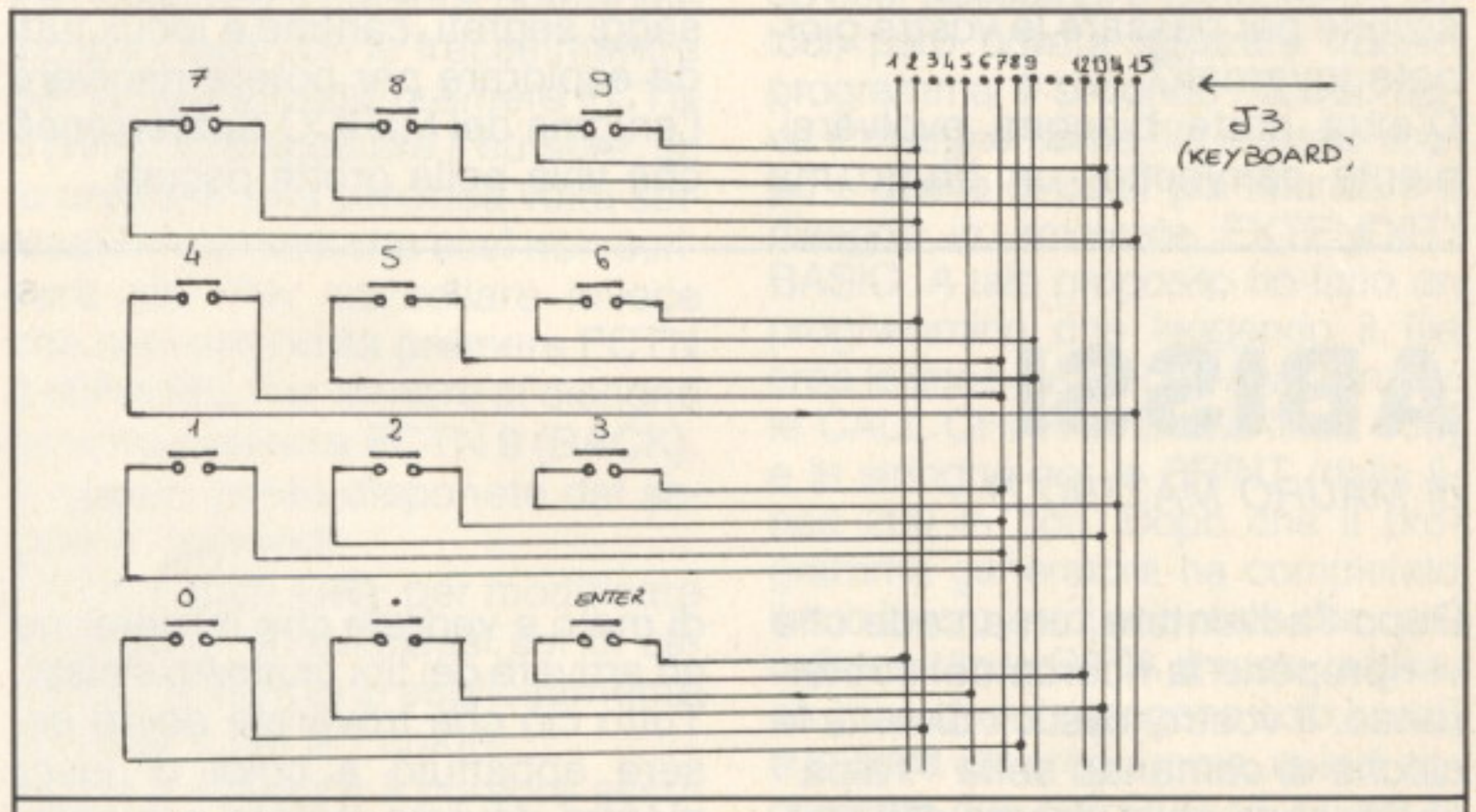
COL / ROW	P8	P13	P14	P15	P9	P12
P7	1	2	3	4	5	FCTN
P2	ø	9	8	7	6	
P10	Q	W	E	R	T	CTRL
P1	P	O	I	U	Y	ENTER
P3	A	S	D	F	G	SHIFT
P4	;	L	K	J	H	SPACE
P11	Z	X	C	V	B	
P5	/	>	<	M	N	=

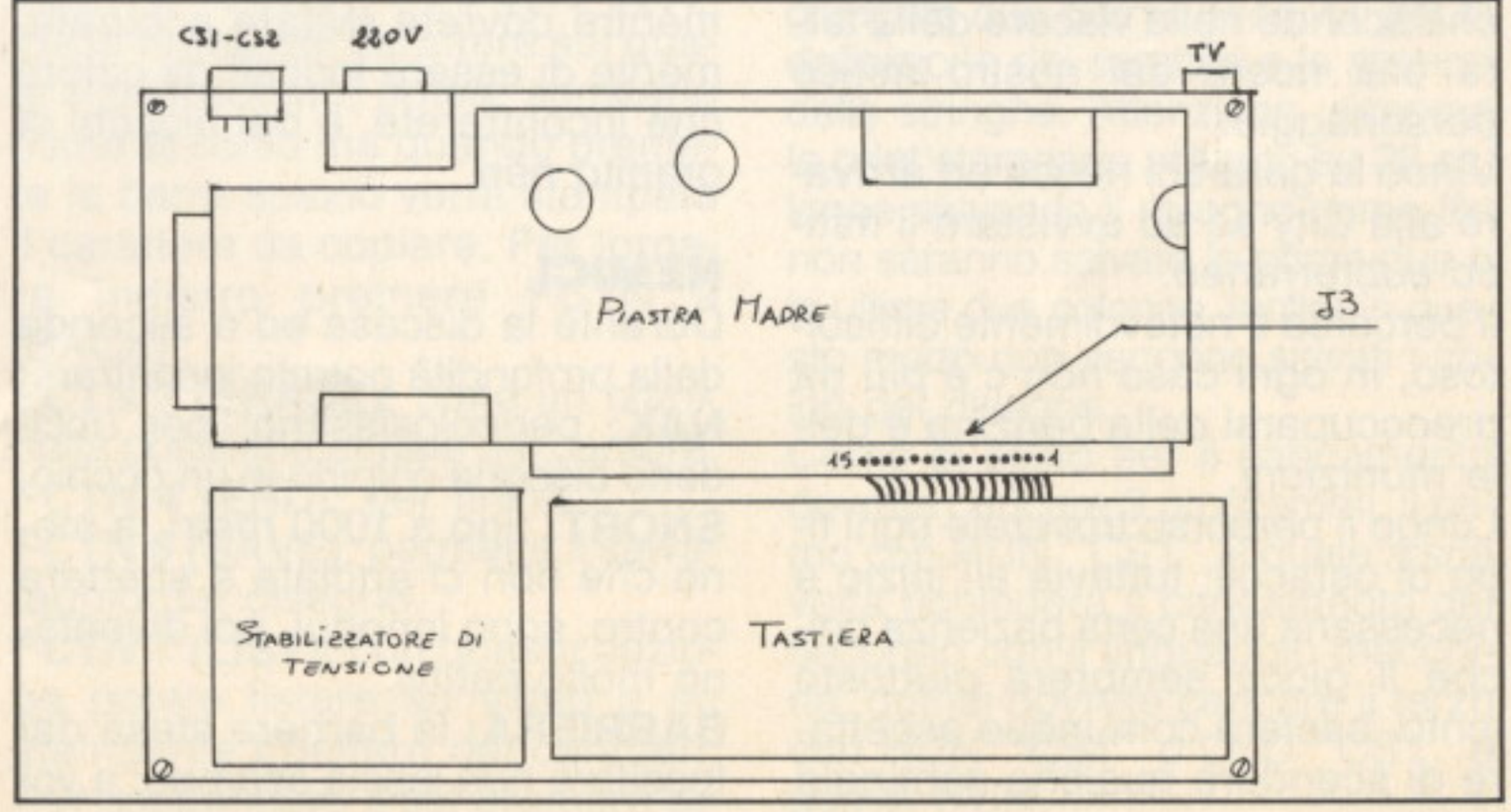


La realizzazione pratica dovrà quindi essere particolarmente curata, anche se eventuali errori, oltre a non causare nessun danno, saranno immediatamente rilevati. Infatti se ad esempio schiacciando il tasto ''3'' verrà visualizzata una ''A'' dovrete avere il sospetto che qualcosa non funziona correttamente.
Una volta terminata la costruzione vedrete che dal vostro TN usciranno dieci fili identificabili per il loro diverso colore.
Saranno questi dieci fili che andranno collegati alla porta J3, ma è com unque possibile installare una presa a jack (10 poli), sul corpo del calcolatore per eseguire i collegamenti necessari su di essa, e poter attaccare e staccare a piacimento il TN.
Passiamo ora alla installazione vera e propria:
È necessario aprire la ''scatola'' che contiene il computer, e per fare ciò si devono togliere le sette viti che sono sotto di essa.
Eseguita questa operazione si deve sollevare il fondo ponendo particolare attenzione all'interruttore che risulta essere un pò difficoltoso da staccare.
Dopo aver tolto la copertura inferiore avrete davanti una situazione del tipo rappresentata in fig. 4, dove è anche indicata la porta J3 dalla quale si dipartono i 15 fili che collegano la tastiera.
A questo punto il collegamento dei nostri dieci fili è affidato al vostro ''buon senso'', in quanto un lavoro arrangiato è sicuramente da sconsigliarsi, ma anche aggirarsi per un computer con un saldatore in mano non è da tutti.
In commercio esistono delle specie di ''doppie spine'' per queste porte, ma il loro reperimento è un po' difficoltoso.
La soluzione più pratica è quella di cercare di collegarsi sulla spina che arriva dalla tastiera facendo attenzione a non cortocircuitare involontariamente dei pin.
Una volta eseguiti i collegamenti si tratterà di portare i fili all'esterno del TI 99 tramite un piccolo foro, oppure attraverso le grigliette che già esistono per il passaggio dell'aria. Naturalmente se avrete optato per la presa sul corpo del calcolatore questo problema non sussisterà.
In caso contrario ricordatevi di far passare i fili prima di eseguire i collegamenti, pena dover smontare tutto e ricominciare.
Anche se questo passo è stato completato eseguite qualche test per verificare il buon funzionamento di TN e passate alla chiusura del TI 99.
Porre nuovamente attenzione all'interruttore, che fa un po' fatica a rientrare nella sua sede, e reinserire le viti.
Ricollegate il calcolatore alla rete, e da questo momento il vostro amato texas avrà 12 tasti in più!!!



Componenti:
— 12 interruttori normalmente aperti (tasti)
— Contenitore plastico (100 × 80 × 40 mm. circa)
— Cavo a 10 poli (50 cm. circa)





GROTTE OSCURE

di MMG

Inverno, tempo di neve, le giornate passano tristemente lente, e di solito, si cerca di passarle in pantofole, al calduccio; a queste parole sembra fare eco una vecchia pubblicità che diceva: ''Voi, il caminetto acceso una calda poltrona, un libro e...'' e qui completiamo ''...Il vostro TI 99''.
Infatti, se una volta, quando non c'erano i computer ci si poteva accontentare di un buon libro d'avventure, oggi c'è la possibilità di passare la giornata immedesimandosi, anzi cercando di risolvere le situazioni che il libro... pardon, il computer ci pone davanti. In effetti quale sistema migliore di un Adventure pensate di avere a disposizione per passare le vostre giornate invernali?
D'altra parte bisogna evolversi, niente caminetto, un brutto ma pratico radiatore, e quindi niente libro, ma un evoluto quanto divertente Adventure.
Tutto questo per presentarvi il gioco che la MMG software ci vuole offrire.
Il gioco prende spirito da diverse fiabe; inizierete in un bosco dove troverete diverse cose e diversi oggetti, starà a voi eseguire le azioni ed impartire gli ordini opportuni. Tra i personaggi che incontrerete ci sono Orchi, Serpenti, Draghi, Cani a tre teste ecc... tutto nella più rigorosa fedeltà della tradizione Tolkeniana.
Anche qui come in Advanced, Dungeons e Dragons ci sono miriadi di stanze da visitare, e passaggi segreti, cantine e loculi tutti da esplorare per potere risolvere l'enigma del MITICO Subterraneo che vive nelle grotte oscure.

IL GIOCO

L'Adventure è composto da due programmi ben distinti, il primo serve per presentare il gioco, elenca i comandi che possono essere usati ed il loro significato.
Dopo la presentazione il 1° programma chiama il caricamento del secondo, dove inizia il gioco vero e proprio. È da notare che il gioco offre diverse possibilità di risoluzione. Il gioco chiaramente non è risolvibile facilmente, e quindi pensiamo che offrirà ore ed ore di divertimento a tutti. Questo Adventure fa parte della serie di Adventure che la MMG ha messo a punto tra cui troviamo: Il Mistero dei Puffi ed il Castello del Vampiro.
Ovviamente i titoli non si fermeranno qui, ed in futuro ne saranno disponibili altri, nel frattempo cercate di risolvere questo di mistero.

ABISSI

di MAURO MATTIAZZI

Dopo l'adventure, un arcade che vi ripropone la ricerca del subterraneo. Il vostro joystick diventa la cloche di comando della ''Talpa'' che scende nelle viscere della terra alla ricera del nostro mitico personaggio.
Vince la gara chi riesce ad arrivare alla City ed ad avvistare il mitico subterraneo.
Il percorso è notevolmente difficoltoso, in ogni caso non c'è più da preoccuparsi della benzina e delle munizioni.
Lungo il percorso troverete ogni tipo di ostacoli, tuttavia all'inizio è necessaria una certa pazienza poiché il gioco sembrerà piuttosto lento, basterà comunque aspettare di scendere qualche centinaio di metri e vedrete che inizieranno ad arrivare dei tipi piuttosto sinistri. Tutto ciò che troverete dovrà essere abbattuto a colpi di laser mentre dovrete evitare accuratamente di essere toccati da coloro che incontrerete, e dai blocchi di granito neri.

NEMICI

Durante la discesa ed a seconda della profondità potrete incontrare:
NAK: pericolosissimo; per ucciderlo bisogna colpirlo in un occhio.
SNORT: fino a 1000 metri, a meno che non ci andiate a sbattere contro, sono innoqui, poi diventano molto cattivi.
BARRIERA: la barriera stesa dal tessitore non lascia scampo: a voi scoprire come passarci in mezzo.
WHYTE: creano una immagine mentale; dovete colpire il mostro giusto altrimenti l'immagine viene subito rigenerata.
SYN: mostri parapsicologi; con la forza mentale sbattono la vostra nave contro le pareti della grotta. Si trova solo alle grandi profondità ma non è escluso che possiate trovarlo anche in superficie.

COMANDI

Per comandare l'astronave si usa il joystick, in alto per fermarsi, in basso per ripartire, destra e sinistra per le direzioni laterali.
Per sparare il pulsante rosso.
Ogni 1000 metri viene assegnata una nuova astronave.

EASY GRAPHICS

di GIANPAOLO BOTTIN

Questo mese vi presentiamo non uno dei soliti giochi ma un potente EDITOR grafico che permette la realizzazione di schermate, utilizzando i 112 caratteri a disposizione dell'EXTENDED BASIC. Per motivi pratici il programma può girare soltanto se si dispone dell'espansione di memoria, però il disegno risultante può essere utilizzato anche con il solo EXTENDED BASIC.

Easy Graphics per il suo funzionamento utilizza una serie di programmi in linguaggio macchina di cui ci occuperemo a parte, per il momento ci basterà sapere che tali programmi permettono la gestione completa di una seconda pagina video la quale, al contrario di quella solita, permette l'uso di tutti i 255 caratteri. Naturalmente se il disegno dovrà essere utilizzato in ambiente BASIC, senza espansione, ci si dovrà limitare ad usare caratteri da 32 a 143.

Dopo il RUN comparirà una schermata di presentazione (naturalmente creata con questo stesso programma) dopo di che premendo un tasto vi verrà chiesto quale carattere intendete utilizzare per lo spazio, cioè per le zone vuote del disegno. Di solito si usa il carattere 32 anche perché è quello utilizzato dal Basic, ma se intendete usate i 255 caratteri a disposizione potete benissimo scegliere il carattere che preferite tenendo presente che il carattere 1 è utilizzato per il pattern del cursore a croce. In seguito vi viene richiesto qual è il primo carattere che volete utilizzare per il disegno. Dopo aver premuto ENTER vi trovate con il cursore sulla pagina che conterrà il vostro disegno, posizionatevi dove volete iniziare a disegnare e premete FCTN 8. Ora potrete lavorare su un ingrandimento del carattere da voi scelto; mentre a sinistra vi è la vostra area di lavoro a destra potete vedere il carattere nelle sue solite dimensioni con attorno gli 8 caratteri che lo circondano nel disegno. Comunque prima di tutto ciò vi viene mostrato in alto quale è il primo carattere disponibile, quindi vi si chiede quale carattere volete usare. Se il carattere su cui vi eravate posizionato era uno spazio vi verrà proposto il primo carattere libero altrimenti vi verrà stampato il codice del carattere su cui era il cursore quando avete premuto FCTN 8. In seguito sarà stampato il set di caratteri a cui appartiene quello da voi selezionato, quindi i colori che potete anche modificare.

Per spostare il cursore potete utilizzare i tasti con le frecce mentre per scrivere basta premere FCTN 5 (WRITE) e spostare il cursore, se lo premete una seconda volta sollevate il cursore che così non scriverà più. Per cancellare invece che scrivere basta premere FCTN 3 (ERASE). Per tornare al disegno occorre premere FCTN 9 (BACK).

A questo punto disponete dei seguenti comandi:

FCTN 1 (COLOR): per modificare il colore di un qualsiasi set di caratteri. Vi vengono esplicitamente richiesti il set e i colori.

FCTN 2 (COPY): il cursore assume la forma del carattere sul quale è posizionato, potete spostarlo come al solito ma quando premete la barra spazio verrà stampato il carattere da copiare. Per tornare indietro premere FCTN 9 (BACK).

FCTN 3 (ERASE): pone un blank nella posizione attuale del cursore.

FCTN 4 (END): per finire.

FCTN 6 (SAVE): permette il salvataggio del disegno.

FCTN 7 (LIST): con questa opzione potete listare su video o su stampante i pattern dei caratteri. Per sospendere il list premere un tasto, per finire la barra spazio.

FCTN 8 (ZOOM): permette di definire i pattern dei singoli caratteri.

FCTN 9 (BACK): ritorno al disegno.

CTRL 1 (SCREEN): modifica il colore dello schermo e del cursore.

CTRL 2 (GRID): fa comparire una griglia che può aiutare nello spostamento del cursore, per farla scomparire premere di nuovo CTRL 2.

CTRL 6 (LOAD): permette di caricare un disegno salvato in precedenza.

Infine tramite i tasti delle frecce più il tasto FCTN potete effettuare uno scrolling del disegno in tutte le direzioni. Una particolarità di questo programma è data dal salvataggio dei disegni. Vi sono due tipi di salvataggio, il primo potete selezionarlo semplicemente rispondendo N alla domanda se volete salvare il disegno come FILE. In questo caso verrà salvata direttamente la memoria VDP che contiene il disegno permettendo così salvataggi e caricamenti veloci, però dovete utilizzare questo programma. Il secondo metodo salva il disegno come file e il suo scopo è quello di poter poi ricaricare il disegno in ambiente EXTENDED BASIC. A tale proposito ho fatto un programmino che leggendo il file crea le linee DATA con i pattern per le CALL CHAR (fino alla linea 160) e le stringhe per le PRINT (dalla linea 200 in poi). Dopo che il programma generatore ha completato il suo lavoro a voi non resta che cancellare le linee DATA rimaste in bianco (cioè che contengono solo spazi) e le linee del programma, quindi aggiungere due cicli FOR-NEXT per la definizione dei caratteri e la stampa delle stringhe. Attenzione, siccome le print stampano soltanto su 28 colonne salvando il disegno come file non saranno salvate le prime due e le ultime due colonne, inoltre in questo modo non vengono salvati i colori del disegno.

Come al solito per il caricamento dovrete caricare 3 programmi, il primo con RUN "CS1", il quale a sua volta richiederà il caricamento del secondo con il linguaggio macchina, quindi dovrete caricare il terzo con RUN "CS1" o OLD "CS1" e quindi RUN.

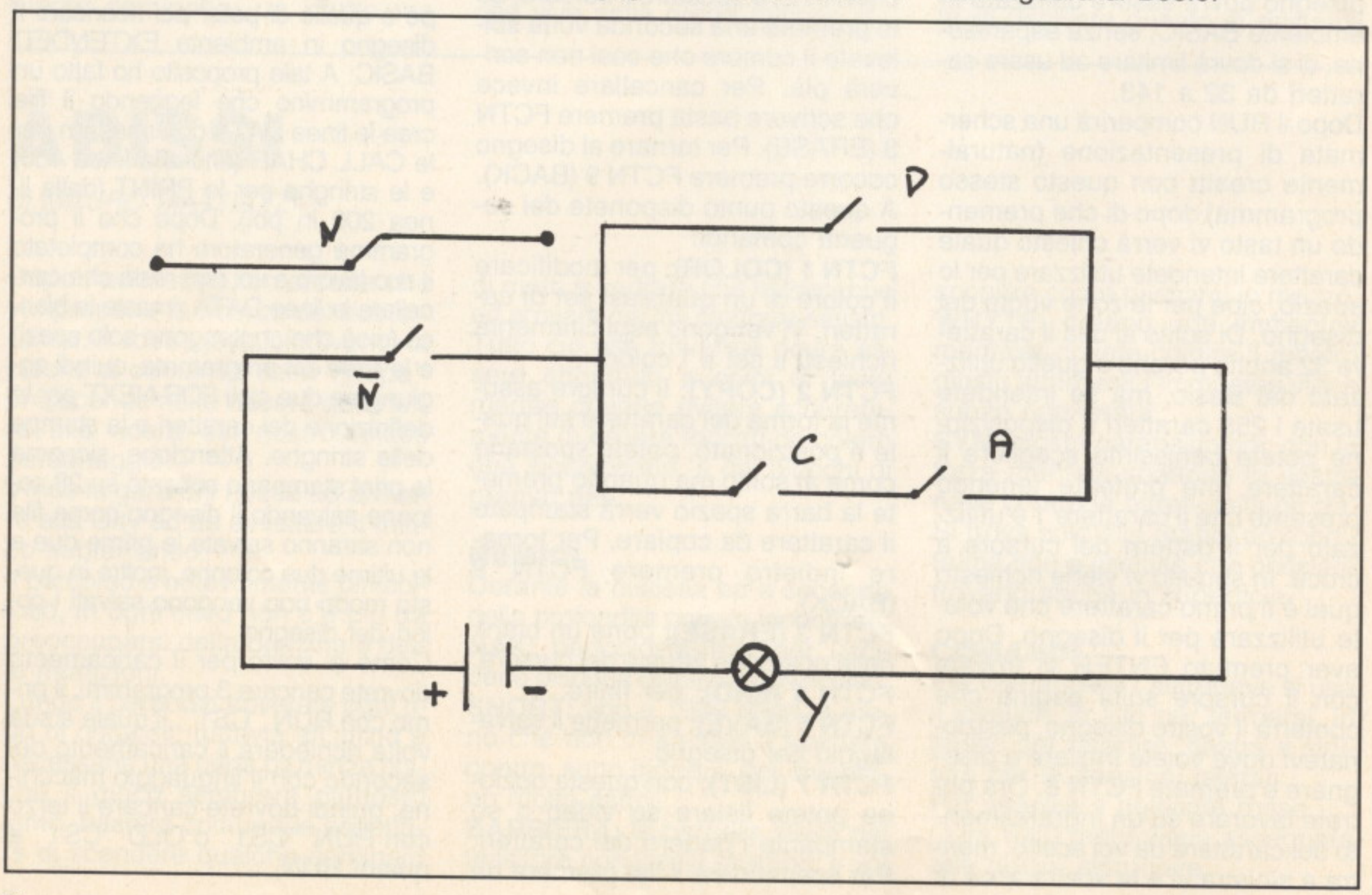
CIRCUITI LOGICI

di FRANCO LATINI

Il programma permette di verificare passo per passo le operazioni logiche fondamentali che sono alla base del funzionamento dei circuiti digitali. Nel caso della funzione booleana è possibile introdurre tutte le 16 alternative agli interruttori, per ciascuna di esse visualizzare lo stato del circuito, vedere la tavola di verità riassuntiva di tutte le alternative fino a quel momento inserite.
Il programma realizzato in TI BASIC, occupa molta della memoria disponibile sia per l'uso abbondante di didascalie, sia a causa della non linearità dei disegni delle porte AND OR, NOT, che hanno richiesto una numerosa quantità di istruzioni di definizione e visualizzazione dei caratteri.

Che cos'è la logica

La logica studia le leggi del pensiero, si fonda sul ragionamento metodico e rigoroso e, al contrario della probabilità, scienza dell'incerto e del probabile, si occupa delle proposizioni, cioè di quelle espressioni verbali che possono assumere due soli valori: quello di verità (V) o quello di falsità (F).
Ad esempio la frase p: ''4 è un numero naturale pari'' è vera (V), mentre la frase q: ''4 è un numero primo'' è falsa (F).
Invece la frase r; ''forse domani nevicherà'' è di tipo probabilistico poiché non si può dire con certezza se essa è vera o falsa.
Una logica a due valori si applica oltre che alle proposizioni anche a molti congegni: un interruttore - o un rubinetto -, possono essere chiusi o aperti; un nucleo di ferrite può essere magnetizzato in un verso o in quello contrario; un diodo o un transistore può avere due diversi livelli di potenziale ai suoi terminali.
Alle proposizioni e a siffatti congegni si applicano alcune operazioni logiche, che pur con diverse denominazioni, portano ad analoghi risultati: le operazioni di congiunzione ($\wedge$), disgiunzione (V), e negazione ($\neg$), della logica proposizionale, sono rispettivamente analoghe a quelle di intersezione ($\cap$), unione (U), complementazione ($\setminus$) della teoria degli insiemi e a quelle di prodotto logico (·), somma logica (+), inversione (—) dell'algebra di Boole.



Questa a differenza dell'algebra ordinaria che tratta variabili e funzioni che possono assumere valori qualsiasi, opera su variabili a due valori: 0 oppure 1.
Essa quindi è di grande aiuto nella progettazione e semplificazione di circuiti digitali i cui diodi possono assumere due e due soli stati. Questi circuiti anzi, sono da ritenersi la realizzazione pratica della loro corrispondente funzione booleana.
Un esempio che evidenzia l'analogia tra le operazioni logiche applicate, alle proposizioni e quelle applicate ai circuiti a relè o a diodi è il seguente:

Si consideri la frase r: ''la cassaforte può essere aperta non di notte e con la chiave del direttore o con le chiavi dell'amministratore e del cassiere''.
Questa affermazione può considerarsi formata dalle frasi:

s: ''la cassaforte può essere aperta''.

p: ''se verrà usata la chiave del direttore''.

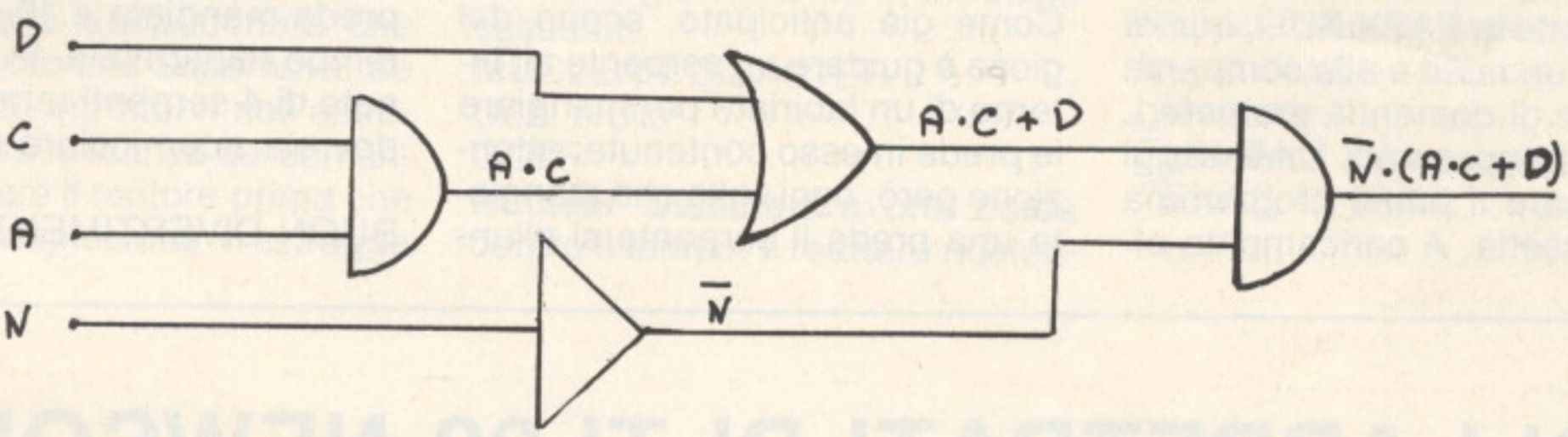
q: ''se verrà usata la chiave dell'amministratore e la chiave del cassiere''.

h: ''non di notte''.

I valori di verità della frase s sono calcolabili con le operazioni di congiunzione, disgiunzione, negazione sulle frasi p,q,n.
Questa stessa situazione è descrivibile con un circuito a relè che realizza la funzione booleana:

$$Y = \bar{N} \cdot (A \cdot C + D)$$

o del circuito a simboli industriali (figura 2).
La tavola di verità individua la 24 = 16 alternative possibili ai contatti D,C,A,N, i valori delle operazioni logiche intermedie e finale, il valore della funzione booleana in corrispondenza a ciascuna alternativa (figura 3).



D	C	A	N	$\bar{N}$	A·C	A·C+D	$\bar{N}\cdot(A\cdot C+D)$
0	0	0	0	1	0	0	0
1	1	1	1	0	1	1	0
1	0	1	0	1	0	1	1
0	1	0	1	0	0	0	0
0	0	1	1	0	0	0	0
1	1	0	0	1	0	1	1
0	0	0	1	0	0	0	0
1	0	0	0	1	0	1	1
0	0	1	0	1	0	0	0
1	0	0	1	0	0	1	0
0	1	1	0	1	1	1	1
0	1	0	0	1	0	0	0
1	0	1	1	0	0	1	0
1	1	1	0	1	1	1	1
0	1	1	1	0	1	1	0
1	1	0	1	0	0	1	0

SNAKE-SNAKE

di BOTTIN GIANPAOLO e MATTIAZZI MAURO

In questo gioco siete un serpente intrappolato in un labirinto e dovete muovervi al suo interno per mangiare tutte le prede facendo però attenzione a non mangiarvi la coda.

Caricamento:
Dopo aver acceso il computer con inserito il modulo MINIMEMORY selezionate l'opzione 2 del menù cioè entrate in EASY BUG, quindi premete un tasto e alla comparsa del punto di domanda premete L compariranno quindi i messaggi per caricare il primo programma della cassetta. A caricamento effettuato comparirà nuovamente il punto interrogativo, ora premete QUIT (FCTN +) ed entrate in TI BASIC, digitate quindi:

CALL LINK (''LOAD'')

e caricate il secondo e definitivo programma che alla fine del caricamento partirà automaticamente.

IL GIOCO

Come già anticipato, scopo del gioco è guidare un serpente all'interno di un labirinto per mangiare le prede in esso contenute; attenzione però, ogni volta che mangiate una preda il serpente si allunga rendendo così più difficili le vostre manovre.
Vi sono sei schermi diversi di difficoltà crescente, inoltre, con il succedersi degli schermi la velocità del serpente aumenta notevolmente.
Ogni schermo dovrà essere completato prima che il tempo arrivi a zero pena la perdita di un serpente. Guadagnate 100 punti per ogni preda mangiata e 10 per unità di tempo risparmiata. In tutto disponete di 4 serpenti terminati i quali dovrete ricominciare da capo.

BUON DIVERTIMENTO.

GLI ARRETRATI DI TI-99 NEWSOFT

Se avete perso qualche numero, o se avete scoperto da poco
la nostra rivista, approfittate delle eccezionali condizioni
che TI-99 Newsoft vi offre per recuperare i numeri che vi mancano.

1 numero arretrato L. 15.000
4 numeri arretrati L. 40.000
10 numeri arretrati L. 80.000

Per le richieste utilizzate l'apposita cartolina inserita nella rivista.

FUN-PAC 2

FUN-PAC: un divertente pacchetto di programmi, uniti in un'unica cassetta.
Questa volta i giochi sono:
ESCAPE THE MUGGER,
STARSHIP SUPERNOVA
GUNFIGHT

Escape the Mugger: è una fuga serratissima contro un rapinatore che vi segue all'interno di una stanza di cui solo lui possiede la chiave. Se tra i vari oggetti che compongono il suo bottino, nell'inseguirvi perde anche la chiave forse potrete farcela ad arrivare alla porta e scappare.

Starship Supernova: è il nome di una navicella abbandonata che precipita verso una Supernova. Se riuscirete ad introdurvi nell'astronave, trovare la stanza dei comandi e spegnere il motore prima che la stella in esplosione vi attragga potrete sperare di tornare sulla terra con un ricco bottino: un'astronave vale sempre qualche lira, e, come in mare, quello che si recupera è proprio.

Gunfighter: è invece un singolare duello tra pistoleri. Dovrete riuscire a essere la ''più veloce pistola del West'' contro cinque abili tiratori.
È un gioco che nella variante a più giocatori permette di impugnare la pistola uno contro l'altro.

FUN-PAC 3

Continua con questa cassetta, FUN-PAC 3, la felice combinazione di tre giochi riuniti in un unica cassetta.
MUCLEAR DISASTER
SEA WOLF
GOLF

Nuclear Disaster: è una corsa contro il tempo: il reattore nucleare si è guastato e lascia fuoriuscire a poco a poco le radiazioni. Gli scienziati presenti al momento dell'incidente, sono svenuti e rischiano di morire se il robot che voi azionate dall'esterno non riuscirà a sospingerli nella camera di decontaminazione prima che il livello radioattivo raggiunga il 100 per cento.
Il gioco **Sea Wolf**, metterà duramente alla prova le vostre abilità di comandante.
Dal radar del vostro sottomarino, avete un quadro chiaro di ciò che vi circonda e potete controllare l'arrivo di qualunque nave nemica.

Golf, è invece una raffinata simulazione sportiva che permette più di una ''finezza'' da gran giocatore.
FUN-PAC 3 è compatibile con il TEXAS TI 99/4A.

A causa di un errore da parte della ditta fornitrice, una partita di cassette risulta mancante di alcune linee dell'ultimo programma del lato 2; per riparare all'inconveniente, ripubblichiamo su questa cassetta, alla fine del lato 2, il programma ''SEA WAR''. Vi preghiamo di scusarci per il disagio arrecatoVi.

La Redazione

TI 99 NEWSOFT

L. 10.000

Numero 15 - Anno III° - Gennaio 1986

MINI MEMORY:
FLASH

HARDWARE:
RESET E LOAD

EXTENDED:
PROFONDO ROSSO

SOMMARIO

☐ **TI BASIC**
Adventureland — pag. 3

☐ **HARDWARE**
Reset e Load — pag. 4

☐ **EXTENDED**
Profondo rosso — pag. 5

☐ **MINI MEMORY**
Flash — pag. 6

☐ **EXTENDED**
Spider — pag. 7

☐ **EXTENDED**
Flyght Simulator — pag. 8

☐ **TI BASIC**
Eusebio
L'ultimo fiammifero — pag. 9

☐ **PICCOLI ANNUNCI** — pag. 10

☐ **comPIUmarket** — pag. 12

SOFTWARE & SOFTWARE **(LATO 1)** **N. 15**

TI BASIC:
☐ L'ultimo fiammifero
☐ Adventureland
☐ Eusebio

MINI MEMORY:
☐ Flash

SOFTWARE & SOFTWARE **(LATO 2)** **N. 15**

EXTENDED BASIC
☐ Spider
☐ Flyght simulator
☐ Profondo rosso
☐ Generatore linee data
per "Easy Graphics"

TI 99 NEWSOFT

ANNO III° - Numero QUINDICI
MENSILE

Direttore responsabile:
ANTONIO C. LOSITO
Coordinatore Editoriale:
VINICIO CIVININI
Direttore Amministrativo:
ENRICO CALLERIO
Art Director:
BEPPE RE-FRASCHINI
Illustrazioni:
MARIA TERESA COCOZZA
Segreteria:
CRISTINA FRISON
Redazione:
FIORENZA AURIEMMA
Collaboratori:
SERGIO BORSANI - GIANPAOLO BOTTIN - DANIELE CATALFAMO - PAOLO CIVARDI - STEFANO DELLI PONTI - FRANCESCO MACALI - MAURO MATTIAZZI - EZIO MONTINI - LEVIO PEROTTI - RICCARDO ROSSI - MARCO SQUINTANI - ALBERTO STRAFILE - SEBASTIANO TOMASELLO

TI 99 NEWSOFT è una pubblicazione della **NEWSOFT snc.**
Via Stefano Jacini, 4 - 20121 MI
Tel. 02/807464 - Telex 324284
REGISTRAZIONE AL TRIBUNALE DI MILANO N. 323 - DEL 7.7.84

© COPYRIGHT NEWSOFT snc

TUTTI I DIRITTI RISERVATI
Manoscritti e foto originali, anche se non pubblicati, non si restituiscono ed è vietata la riproduzione anche parziale di testi, fotografie e listati.

Abbonamenti: l'abbonamento annuo a 11 numeri di **TI 99 NEWSOFT** costa L. 95.000 per l'Italia; L. 160.000 per l'Europa (spedizione aerea); L. 200.000 per tutti gli altri paesi. L'importo, per assegno non trasferibile o per vaglia postale, va inviato a **NEWSOFT snc,**
Via S. Jacini, 4 - 20121 Milano

Stampa: LALITOTIPO srl - Milano

L'editore non si assume responsabilità alcuna sul contenuto degli articoli e/o software inviati. Per gli articoli firmati e/o siglati da collaboratori esterni, la redazione si assume la responsabilità prevista dalle leggi sulla stampa.

Tutti possono collaborare a TI 99 NEWSOFT, sia con articoli che con programmi. Scriveteci o telefonateci in redazione per avere tutti i ragguagli.

ADVENTURLAND

di NORBERTO PEROTTI

Siete un esploratore, e vi trovate in una landa sperduta, piena di insidie, alla ricerca dell'ambitissima coppa d'oro custodita dagli indigeni nel Tempio Maledetto. Il vostro più acerrimo nemico è il dottor Elvin, il quale, venuto a conoscenza della vostra missione, vi ha pedinato e rubato la mappa (per fortuna in tasca ne avete una copia). Elvin è deciso ad arrivare per primo, e cercherà di fermarvi con ogni mezzo. Per ritardare la vostra marcia ha disseminato per la strada spade d'argento e sacchi di denaro; ma non fatevi distogliere: sappiate che ogni volta che prenderete un oggetto, ritornerete al punto di partenza (si intende del quadro dove siete arrivati, non del viaggio).

Primo quadro:
Davanti a voi c'è l'imboccatura della strada, protetta dagli Occhi Malefici. Questi, scendendo, cercano di travolgervi, ma voi potete aprirvi un varco scoccando le vostre frecce. Con il tasto ''F'' precipitatevi poi verso l'imboccatura il più presto possibile e... buon viaggio! N.B. Questo è l'unico quadro con un limite di tempo.

Secondo quadro:
Elvin si è accorto della vostra presenza, e dal suo elicottero scaglia contro di voi tre Occhi Malefici ad ogni passo che fate od ogni dardo che scagliate per liberarvi la strada.

Terzo quadro:
Terribilmente irritato per non essere riuscito ad uccidervi con il bombardamento, Elvin si appresta a minare un campo, ma non sa che voi siete nascosti dietro un cespuglio: potete così osservarlo indisturbati. Un suono vi avvertirà che Elvin ha finito di piazzare le mine; da questo momento potete iniziare a memorizzare lo schema. Dopo pochi secondi Elvin coprirà le mine, e per raggiungere la sommità del video, dovrete basarvi unicamente sul ricordo. Attenti a dove mettete i piedi!

Quarto quadro:
Siete giunti al Tempio Maledetto. Questo tempio ha due porte: una conduce alla stanza dove è custodita la coppa, l'altra alla morte. A voi la scelta. P.S. Elvin non vi darà più fastidio: ha sbagliato porta!!!

Quinto quadro:
Ora potete finalmente vedere la coppa! Fra lei e voi c'è di mezzo un solo ostacolo: il sistema di difesa approntato dagli indigeni. Ecco in cosa consiste: immobilizzati dallo sguardo di 4 occhi, due a destra e due a sinistra, dovete cercare di fermare sul blu il colore della coppa premendo un qualsiasi tasto al momento opportuno. Se vi riuscite, siete liberi di muovervi e di recuperare la coppa; se fallite, ogni occhio scaricherà su di voi un raggio laser, fulminandovi.

Note importanti:
1) Il sacco di denaro e il punto interrogativo vanno toccati solo nell'angolo inferiore sinistro e unicamente con la testa, questo per velocizzare il gioco. 2) Per passare al quadro successivo bisogna sempre giungere alla sommità dello schermo. 3) La spada va toccata nel punto inferiore sempre con la testa. 4) Urtando contro un albero, un occhio, la parete, ecc. si perde la partita.

Tavola dei punteggi

Sacco di denaro	90 pts
Spada	50 pts
Coppa	5000 pts
Quadro	1000 pts
Punto int.	tra 10 e 90 pts
Occhio	100 pts

100- 510	Definizione caratteri
520- 550	Attribuzione colori
560- 780	Presentazione
790- 910	Subroutine: disegno della cornice
920-1070	Lista data generale
1070-1210	Subroutine: legge e visualizza i dati
1080-1270	Visualizzazione primo quadro
1280-1330	Subroutine: posiziona l'uomo sul video
1340-1660	Subroutine: identificazione tasto premuto e branches
1670-1830	Primo caso: urtato sacco di denaro
1840-1890	Subroutine: cancellazione oggetto
1900-2020	Secondo caso: urtata spada
2030-2190	Terzo caso: urtato punto interrogativo
2200-2350	Routine di sparo
2360-2600	Logica del primo quadro
2610-2670	Subroutine: quadro terminato
2680-2740	Visualizzazione secondo quadro
2750-2930	Logica secondo quadro
2940-3160	Visualizzazione e logica del terzo quadro
3170-3450	Visualizzazione quarto quadro
3460-3680	Logica del quarto quadro
3690-4060	Fine gioco e visualizzazione punteggio
4070-4200	Visualizzazione quinto quadro
4210-4260	Logica quinto quadro
4270-4380	Primo caso: errore
4390-4560	Secondo caso: vittoria. Reinizializzazione variabili e branch.

RESET E LOAD

di SEBASTIANO TOMMASELLO

Un piccolo circuitino vi permetterà di avere due nuovi comandi tramite pulsante. È un lavoro semplicissimo, che chiunque sappia utilizzare un saldatore è sicuramente in grado di realizzare. Il materiale occorrente è:
1) tre pezzi di filo lunghi 30 cm circa;
2) una resistenza da 56K;
3) un condensatore da 10nF;
4) un deviatore con ritorno nella posizione centrale (oppure 2 pulsanti).
Vediamo prima di tutto cosa ci consentono di fare questi due comandi.
RESET e LOAD sono *interrupt hardware,* cioè consentono tramite il cambiamento di stato di un piedino del microprocessore, di interrompere drasticamente il lavoro in esecuzione per eseguire un programma il cui indirizzo di partenza è contenuto in una specifica locazione di memoria.
RESET è il comando che viene eseguito ogni volta che si accende la consolle, e che termina con la presentazione del ''MASTER TITLE SCREEN'', praticamente effettua l'inizializzazione del sistema, verificando le eventuali periferiche collegate, caricando il set dei caratteri ecc.
Fin qui, niente di nuovo, senonché normalmente il reset può essere fatto solo spegnendo e riaccendendo la consolle. Con questo circuitino, sarà possibile effettuare il reset del computer in qualsiasi momento, senza spegnere la consolle. Il vettore di partenza è a >0000 per il workspace e a >0002 per il program counter.
LOAD può essere utilizzato solo con l'espansione di memoria. È molto utile perché il vettore di partenza è a >FFFC per il workspace e a >FFFE per il program counter, quindi negli ultimi quattro bytes della ram utente, ciò significa che è possibile far partire qualsiasi programma semplicemente presettando il suo indirizzo di partenza ed il suo workspace nei vettori ''load'' e premendo il pulsante load.
Il programma ''DEBUG'' della Texas è predisposto per funzionare con il load, basta caricare il programma normalmente e mandarlo in esecuzione dopodiché si può tornare al master title screen e far partire il debug premendo il load.

Descrizione del Circuito

Il circuito è semplicissimo: per attivare i reset viene portato a livello basso il piedino omonimo del generatore di clock tramite il pulsante, per attivare il load viene portato a livello basso il pin load del microprocessore ma in questo caso si rende necessario un circuito antirimbalzo per il contatto del pulsante, perché altrimenti il microprocessore eseguirebbe più volte questa operazione. Un circuito che svolga bene questo compito, necessita di qualche integrato, ma nel nostro caso, essendo il circuito molto semplice, ci accontentiamo di una resistenza e di un condensatore. Il condensatore quando viene chiuso il contatto del load fa arrivare solo un impulso al microprocessore. Quando il contatto load torna aperto, la resistenza scarica il condensatore che sarà pronto per poter mandare un altro impulso. Come vedrete questa è un ottima soluzione, l'unico svantaggio è che non essendo l'impulso sincronizzato con il microprocessore, potrà talvolta essere necessario premere due volte il pulsante per abilitare effettivamente il load.

IL CIRCUITO ELETTRICO

Descrizione del montaggio

Ciò che c'è di più complesso in questa realizzazione è la fase di smontaggio e di rimontaggio del computer, per questo motivo verrà descritta dettagliatamente:
1) togliere tutti i connettori e qualsiasi modulo dal computer, capovolgerlo e togliere le viti.
2) muovere leggermente il pannello di plastica e togliere l'interruttore d'accensione.
3) togliere definitivamente il pannello di plastica.
4) togliere le tre viti che tengono fissata la scheda principale del computer, riconoscibile perché completamente schermata.
5) rimuovere la scheda con la schermatura, delicatamente, scollegare i connettori dell'alimentatore e della tastiera.
Questa operazione è abbastanza complicata, la scheda potrebbe rifiutarsi di uscire per via della vicinanza dell'alimentatore, per il connettore della tastiera e per il connettore della porta moduli. In casi disperati sarà necessario smontare anche l'alimentatore (la scheda quadrata) togliendo le due viti che lo fissano.
6) togliere il supporto in plastica dello sportellino della porta moduli (due viti).
7) procedere con la foratura della plastica per l'installazione del deviatore. La posizione ideale è vicino al connettore d'alimentazione. I pulsanti o il deviatore, dovranno essere del tipo in miniatura.
8) togliere la schermatura della scheda principale, ci sono tre viti con relativo dado e due mollette. Fare attenzione a non toccare con le dita i componenti della scheda, le cariche elettrostatiche possono danneggiarli.
9) saldare i tre fili. Il filo che sarà saldato al piedino centrale del deviatore è collegato alla massa, il filo del reset è saldato sul piedi-

no di un condensatore elettrolitico, mentre il filo del load è saldato sul piedino di una resistenza. Quest'ultima saldatura, va fatta con molta attenzione perché è facile creare qualche cortocircuito tra le piste nelle immediate vicinanze. È consigliabile usare colori diversi per i tre fili, per poterli riconoscere anche a schermatura montata.
10) ripristinare la schermatura sulla scheda principale, facendo uscire i fili della fessura.
11) saldare i piedini della resistenza con quelli del condensatore in modo che siano collegati in parallelo (vedi circuito elettrico).
12) saldare il filo collegato a massa al pin centrale del deviatore, il filo collegato al reset ad uno dei pin laterali ed il filo collegato al load ai due componenti saldati fra loro in precedenza; l'altro capo dei componenti va saldato all'altro capo del deviatore.
13) procedere ora all'installazione del deviatore nel pannello posteriore del computer e ripercorrere le operazioni fatte in precedenza per rimontare il computer.

Collaudo

Dopo avere effettuato scrupolosamente tutte le operazioni descritte si può verificare se il circuito funziona. Accendete espansione di memoria e computer, andate in basic e premete reset, il master title screen dovrà ricomparire. Andate ora in editor assembler (o ext. basic o minimemory) e caricate il debug e fatelo girare. Premete reset e di nuovo apparirà il master title screen, provate quindi a premere il load, se tutto funziona correttamente vedrete apparire il debug. Se il computer una volta rimontato non dà più cenni di vita, avete senz'altro rimontato male la schermatura della scheda principale, se invece sono le funzioni del deviatore a non funzionare controllate le saldature. ■

PROFONDO ROSSO

di CARLO RANDONE

Esegue la colonna sonora dell'omonimo film. Il programma è caratterizzato dal fatto che il suono non viene prodotto come serie di note scollegate fra di loro (come capita di ascoltare nella maggioranza dei brani musicali scritti per TI-99!).
La musica risultante è più melodica in quanto non sono state utilizzate direttamente le ''solite'' (!) CALL SOUND, ma ad ogni CALL SOUND è stata sostituita una chiamata ad una apposita procedura (S2 od S3), atta appunto ad eseguire suoni più armonici e gradevoli.
Le procedure S2 ed S3 differiscono fra di loro per il semplice fatto che mentre S2 suona due note contemporaneamente, S3 ne suona tre.
I parametri passati a queste due procedure sono, nell'ordine, i seguenti:
DURATA (misurata in unità convenzionali, in quanto tale valore viene utilizzato come estremo superiore di un ciclo FOR-NEXT di controllo della durata della nota. I valori di questo parametro sono stati fissati in relazione alla velocità dell'interprete basic).
FREQUENZA (espressa in hertz, come al solito)
VOLUME (numero fra 0 e 30, come di consueto).
Seguono ripetizione di frequenza e volume per la seconda ed (eventualmente) terza nota.
Il ''trucco'' sta nel fare sempre delle CALL SOUND con durata molto elevata, ma negativa, e far seguire a tali CALL SOUND un ciclo FOR-NEXT di controllo della durata. Ricordiamo, infatti, che durate negative significano, per il TI 99, che la nota suonata deve essere immediatamente interrotta qualora ne succeda un'altra!
Il programma tende a rifarsi alla esecuzione originale su disco, ridotta a tre voci, tante quante il TI 99 permette.
Anche la timbrica limitata del sintetizzatore TMS 9919 può rendere monotona l'esecuzione. Infatti il brano si sviluppa riprendendo in modo relativamente limitato (come numero), pochi temi con varie timbriche. Questa caratteristica della versione originale non è stato possibile implementarla sul TI 99-4A. Qui i temi, raccolti in sette blocchi ripetutamente chiamati, sono eseguiti sempre con lo stesso timbro (tipo ''organo elettronico'', per intenderci). ■

ANALISI DEL LISTATO

90	intestazione
160	chiamate alle varie parti del brano musicale
260	primo blocco
580	secondo blocco
900	terzo blocco
1100	quarto blocco
1160	quinto blocco
1780	sesto blocco
2520	settimo blocco
2700	procedura S2. Suona due note contemporaneamente nel modo precedentemente descritto.
2710	procedura S3. Suona tre note contemporaneamente nel modo precedentemente descritto.

FLASH

di FABIO TARTAGLIONE

La routine, di nome ''FLASH'', permette di generare l'effetto intermittenza, funzione presente su molti personal ma assente sul TI 99, la frequenza della quale può essere assegnata a piacere, inoltre è prevista una routine di scansione tastiera che restituisce il controllo all'interprete basic solo alla avvenuta pressione di un tasto il cui codice è associato ad una variabile di ritorno. La sintassi di chiamata è la seguente:
CALL LINK (''FLASH'' A,B,C)
dove:
A = Variabile o valore compreso tra 1 e 32767 corrispondente al ritardo; valori maggiori provocano frequenze minori.
B = Variabile o valore compreso tra 0 e 5 corrispondente al N° di tastiera secondo quanto specificato nel manuale allegato alla consolle. Non è previsto alcun controllo per eventuali valori overrange.
C = Variabile di ritorno che conterrà il codice ASCII del tasto premuto.
Secondo gli schemi del CP/M indicherò per semplicità il carattere Control F con ^ F.
Per provare la routine caricatela in minimemory con l'opzione L dell'EASYBUG quindi entrate in ambiente Basic.
Provate a battere

^ F1234567890^ F

quindi Enter e sul video comparirà un messaggio di errore ma senza preoccuparci di questo impostate:
CALL LINK (''FLASH'', 1000,0,K)
a questo punto il messaggio inizierà a lampeggiare, per terminare l'effetto premete ad esempio il tasto 1 (uno), l'intermittenza cesserà ed il controllo sarà nuovamente all'interprete basic. La variabile K conterrà il codice ASCII del carattere 1 ovvero 49, per provare provate con una PRINT K.
I messaggi che si vogliono rendere intermittenti possono essere disposti in qualunque posizione dello schermo, purtroppo a causa della bassa velocità di accesso alla memoria VDP esiste un leggero evidente dissincronismo tra messaggi in testa e messaggi in coda alla pagina video; questo effetto di ''sfarfallìo'' può essere ridotto in maniera accettabile stabilendo tempi di ritardo molto elevati quindi di frequenze basse. Il problema non sussiste o comunque non risalta all'occhio nel caso i messaggi siano abbastanza vicini, lascio comunque la scelta agli utilizzatori.
Uno degli impieghi principali penso possa essere nei programmi utilizzanti finestre e maschere o ancora menù per la scelta di operazioni possibili.
Supponiamo di essere nell'ultimo caso citato, in un menù del tipo:

Menù principale
1) inizio lavoro
2) .....
.....
..... Fine lavoro

— Opzione?

Vogliamo che il messaggio ''Menù principale'' e ''Opzione?'' siano intermittenti, il programma dovrà essere modificato come segue:

```
PRINT
''^ FMENÙ PRINCIPALE^ F''
PRINT  ''= = = = = = = = = = =''
PRINT ''1) Inizio lavoro''
PRINT ''2) .....
PRINT ''.....''
PRINT ''..... Fine lavoro''
PRINT ''^ FOpzione?^ F''
CALL LINK (''FLASH'', 10000,0,K)
```

La routine include quindi anche il ciclo di riconoscimento di avvenuta pressione di un tasto restituendone anche il codice per le elaborazioni successive, in questo caso il codice sarà contenuto nella variabile K.

Descrizione del programma

La routine Flash è collocata in Ram a partire dall'indirizzo esadecimale 7D20, necessita per il suo funzionamento di un buffer di 1/0 il quale è collocato all'indirizzo esadecimale 7A00.
Si tenga presente comunque che non è possibile anche spostando la routine ed il buffer evitare di sovrapporsi al programma new che è quindi inutilizzabile.
Per velocizzare al massimo la routine ho impiegato due registri come flag indicatori di stato ed ho ottimizzato tutti i percorsi, non escluso comunque che possa essere ulteriormente velocizzata e compattata. ■

I registri della Workspace area sono utilizzati nel seguente modo:

R0 =	Puntatore VDP RAM
R1 =	Buffer 1/0 dati
R2 =	Puntatore RAM
R3 =	0
R4 =	Flag 1 (0 = Trasferimento dati VDP/RAM RAM/VDP inattivo)
	(1 = Trasferimento dati VDP/RAM RAM/VDP attivo)
R5 =	Flag 2 (0 = Trasferimento dati VDP/RAM - RAM)
	(1 = Trasferimento dati RAM - VDP/RAM)
R6 =	Codice ASCII tasto premuto
R7 =	Ausiliario per loop di ritardo
R8 =	Valore di ritardo passato dalla call link
R9 =	Codice tastiera passato dalla call link
R10 =	Indirizzo di ritorno interprete Basic

SPIDER

di RAFFAELE FERRIGNO

Il programma prevede il tentativo da parte del giocatore, rappresentato da un omino blu visto dall'alto, di uscire da un labirinto di ben cento camere. Alle difficoltà già insite nella suddetta impresa si aggiunge la circostanza che le stanze del labirinto sono infestate da giganteschi ragni il cui morso è letale.

Fortunatamente raggiungendo, con il consueto uso dei tasti con le frecce, il centro della camera, contrassegnato da un quadrato rosso, prima dei pericolosi ragni, si ha la possibilità, rispondendo esattamente al quesito aritmetico che verrà posto, di distruggere il ragno e poter così, con tutta calma, scegliere quale delle uscite, fino a quel momento chiuse, imboccare nel tentativo di raggiungere l'agognata meta.

Un errore nella risposta al quesito oppure il raggiungimento del centro della camera da parte del ragno comportano ovviamente la prematura fine del gioco.

Il programma, dopo la schermata di presentazione e le istruzioni, prevede la possibilità di poter scegliere il livello di difficoltà del gioco. Esso può variare da 1 a 999 e tale numero rappresenta il massimo fattore, scelto a caso, della moltiplicazione il cui secondo fattore, anch'esso determinato casualmente, può invece variare solo da 1 a 10. In caso di errata risposta il computer fornisce comunque quella esatta, ma il gioco, come già detto, ha termine.

Alla conclusione del gioco, sia essa positiva o negativa, è possibile ricominciare, ovviamente dal punto di partenza, ma con la possibilità di cambiare il livello di difficoltà. Si può anche interrompere il programma in qualsiasi momento premendo semplicemente il tasto (+). Per un uso didattico a livello di scuola inferiore dell'obbligo (il programma è stato redatto ad uso e consumo dei figli dell'autore alunni, appunto, delle elementari) si consiglia di limitare il livello di difficoltà al corrispondente livello di apprendimento del piccolo giocatore.

DESCRIZIONE DEL LISTATO

100- 430	Titolo
440- 480	Istruzioni
490- 520	Iniziazione
530- 560	Scelta difficoltà
570- 590	Disegno quadro
600- 640	Definizione luogo
650- 670	Posizionamento omino
680- 700	Posizionamento ragno
710	Posizionamento centro
720- 750	Posizionamento uscite
760- 840	Movimento omino
850- 880	Movimento ragno
890- 990	Coincidenza omino-ragno e fine negativa
1000-1100	Coincidenza omino-centro ed input quesito
1110-1190	Coincidenza omino-uscite e conclusione positiva
1200-1230	Subprogrammi

ELENCO PRINCIPALI VARIABILI

A,B	Direzioni moto ragno
A$,B$,C$	Stringhe fattori e risultato moltiplicazione
CTR	Variabile di tasto nelle CALL KEY
D	Lunghezza del quesito
F	Ultima uscita utilizzata
FLAG	Flag centro da raggiungere
G	Variabile di risposta nelle CALL COINC
H	Carattere ragno
J	Variabile di ciclo
K	Variabile di tasto nelle CALL KEY
L	Luogo in cui ci si trova
LR	Lunghezza della risposta del giocatore
M	Livello di difficoltà
N,E,S,W	Uscite nelle quattro direzioni
Q	Colonna in cui va scritto il quesito
R,C	Coordinate dell'omino
R$	Risposta del giocatore al quesito
ST	Variabile di stato nelle CALL KEY
X,Y	Coordinate del ragno
X1,X2,X3	Fattori e risultato della moltiplicazione

FLYGHT SIMULATOR

di MMG

Il gioco:
Il gioco consiste nel decollare e quindi guidare l'aereo fino all'aereoporto scelto e naturalmente atterrare.
Fino a qui niente di strano o difficile, il difficile viene quando, si gioca. Infatti, i comandi, che sembrano semplici, le 3 variazioni di stato, faranno emergere delle situazioni abbastanza vicino a situazioni reali mettendo a dura prova la vostra capacità di analisi ed il vostro sangue freddo.

I Comandi:
Il cruscotto visualizzato, contiene tutti i parametri utili ad una guida completa: la parte superiore è il display del computer di bordo, che di volta in volta vi chiederà le modifiche di volo oppure vi darà gli allarmi. A sinistra, sotto il display troviamo i FLAP.
Esso indica l'angolo di inclinazione dei timoni delle ali. Se l'inclinazione è negativa l'aereo scende, viceversa sale.
È importante sapere che l'aereo sale di un angolo 3 volte superiore a quello indicato dai flaps. Per volare senza variazioni di quota, è necessario che, il valore riportato nel quadrante sia portato a zero.
Subito dopo troviamo il rilevamento RADAR, che ha riscontro sul quadrante. Quando partirete, al centro del radar, vedrete una specie di freccia, che indica la rotta da seguire, e a lato, un simbolo che sta a indicare la Vs. posizione rispetto la rotta da seguire.
Nel quadro RILEVAMENTO, questo errore assume un valore numerico; quando questo numero è uguale a zero, vedrete la freccia sparire ed il simbolo prendere il suo posto: questo starà ad indicare che siete in rotta.
È da tenere presente che più gironzolate fuori rotta, facendo quindi più strada, e più consumerete il carburante. Il quadro velocità vi terrà aggiornato sulla Vs. velocità.
È importante sapere che non potete decollare se non superate la velocità di 150 miglia, e che a tale velocità mentre siete per aria l'aereo andrà in stallo precipitando di conseguenza, è da notare inoltre che il vento vi frena ad ogni ciclo di 5 miglia, e che più salite, e più l'aereo, per effetto dell'aria, tenderà a rallentare, mentre, viceversa aumenterà la velocità in caso di discesa.
Da tenere d'occhio è anche il RANGE, che indica la distanza che vi resta da percorrere.
Quando il Range diventa negativo, vuol dire che il controllo passa al quadrante PERCORSO, che è l'indicatore della distanza che vi resta da percorrere. Normalmente a questo punto dovreste già essere in procedura di atterraggio, o addirittura in fase di rullaggio.
FUEL stà ad indicare il propellente che vi resta nei serbatoi, occhio alla spia, quando è rossa vi segnala che siete in riserva e di conseguenza vi resta ben poca strada da percorrere. Il quadrante ''ACC'' vi segnala sempre l'accelerazione e la decelerazione. Il quadrante ''T'', oltre a darvi l'orizzonte vi segna l'inclinazione del timone, anche qui come per i flaps abbiamo una inclinazione positiva e negativa.
Il timone serve per portare in rotta l'aereo, se il valore del timone è zero, l'aereo viaggia diritto, altrimenti ad ogni ciclo comandi l'aereo sterza di tre volte l'angolo del timone.
L'altitudine viene visualizzata ed aggiornata continuamente. Ricordate che i carrelli devono essere fuori per altitudini non superiori alle 50 miglia. I carrelli sono fuori quando la terza spia è colorata di giallo. Per alzare o abbassare i carrelli premere la ''C''.

Il ciclo comandi
Per semplicità, l'aereo si pilota con il ciclo comandi: il computer di bordo vi chiederà: — vuoi variare la velocità? +75/—75—; se volete intervenire, potete aumentarla o diminuirla, digitando *obbligatoriamente* il segno (''—'' per frenare, '' + '' per accellerare). Se la velocità vi stà bene digitate ''0'' sempre però preceduto da un segno.
Dopo avere aggiornato gli strumenti vi chiederà se volete variare il timone, anche qui valgono le stesse regole di prima. Infine vi chiederà se volete variare i flaps per alzarvi od abbassarvi.
Come vedete i comandi sono facili e pochi ma non è facile variare il parametro voluto calibrando l'uno o l'altro comando.

Missione compiuta!
L'atterraggio, per avere esito positivo deve essere effettuato portandosi ad ALTEZZA 0, PERCORSO 0, VELOCITÀ min. 10, CARRELLI fuori, RILEVAMENTO 0.
Qualora una di queste condizioni non fosse soddisfatta il computer vi informerà sulle conseguenze.
A disposizione avete ben 10 aereoporti diversi da scegliere, ed ogni aereoporto ha difficoltà diverse, vi possiamo assicurare che con questo gioco passerete ore molto divertenti, e... occhio ai vuoti d'aria.

EUSEBIO

di MARCO OTTAVIANI

Eusebio è riuscito ad introdursi furtivamente nel castello nemico; la sua missione è quella di aprire le porte del castello, operazione che ha già compiuto, e quindi dare il segnale di attacco alle sue truppe, suonando per tre volte la campana della torre del castello.
Per arrivare alla torre però, deve uscire allo scoperto e saltare da un merlo all'altro per tutta la lunghezza del bastione. Eusebio sa che appena si muoverà una guardia dal basso noterà la sua presenza e comincerà a scalare il muro per raggiungerlo. Se la guardia riesce a raggiungere la sommità delle mura prima che Eusebio sia riuscito a ripararsi dentro la torre, scaglierà una freccia che infallibilmente colpirà il nostro eroe.
Il nostro Eusebio dovrà quindi dimostrare velocità nello spostarsi (con le frecce S&D) e nel saltare (con la barra spazio), ma soprattutto nervi saldi, perché basta il più piccolo errore per precipitare dalle mura. Il gioco è molto semplice in apparenza, ma non fatevi trarre in inganno dall'entusiasmo o dalla fretta; infatti, per compiere la sua missione, Eusebio dovrà percorrere dieci quadri senza il minimo errore, il che richiede una buona dose di freddezza e precisione nell'esecuzione. Potete inoltre impostare la difficoltà del gioco, scegliendo fra tre diversi livelli. Buon divertimento! ■

L'ULTIMO FIAMMIFERO

di SACCHI & FERMANTI

Questo gioco appartiene alla famiglia dei ''Nim'', giochi di strategia nei quali due avversari si fronteggiano sottraendo da un certo numero iniziale di oggetti (generalmente rappresentati da fiammiferi) alternativamente delle quantità opportune allo scopo di evitare la presa dell'ultimo, che sancisce la perdita del gioco.
Qui in particolare, la prima mossa comporta obbligatamente l'eliminazione di un solo fiammifero mentre nei turni successivi ogni giocatore può toglierne un numero a piacere tra uno e il doppio di quelli prelevati dall'avversario nella mossa precedente.
L'avversario con cui ci si deve confrontare è il computer che dà l'impressione di essere imbattibile; armandosi però di pazienza e sagacia si può constringerlo alla resa, cosa che sportivamente riconoscerà.
A dispetto delle regole molto semplici, il gioco è appassionante e stimola la ricerca delle mosse vincenti; una volta raggiunta l'assuefazione si ha la possibilità di modificare le condizioni di gioco variando il numero iniziale di fiammiferi (che tipicamente è di ventuno) da 5 a 39, e anche l'ordine di partenza. Trattandosi di una conversione in Ti Basic di un programma per Olivetti M 20, è necessariamente previsto il frequente salto a routine di ''Print at'' e ''Accept at'' collocate in fondo al programma, che però non comportano alcun rallentamento nello svolgimento del gioco. ■

100- 190	Presentazione del gioco
200- 840	Inizializzazione e dimensionamento variabili, richieste num. iniz. fiammiferi e di chi inizia, controlli accettabilità risposte.
850-1100	Schermata del quadro di gioco
1110-1230	Prima mossa e aggiornamenti situazione
1250-1280	Seconda mossa, controlli fine gioco, test separazione mossa giocatore-computer
1300-1380	Terza mossa e successive
1390-1560	Domanda altra partita e fine gioco
1570-1640	Routine corretto posizionamento numero fiammiferi
1650-2170	Mossa calcolatore, modifiche del quadro, controllo fine gioco
2180-2450	Mossa sfidante, controllo compatibilità, modifica del quadro, controllo fine gioco
2460-2520	Routine print at
2530-2670	Routine accept at
2740-2880	Routine cancella fiammiferi

NOVITÀ DEL MESE

TI-99 comPIUmarket

Per far fronte alle sempre più numerose richieste che ci arrivano, *TI-99 comPIUmarket* amplia il suo catalogo di vendita per corrispondenza.
TI-99 comPIUmarket ricerca e seleziona per voi tutto il software e l'hardware disponibile per il TI-99/4A, nonché tutti gli accessori necessari alla gestione e alla manutenzione del vostro computer proponendovi nuovi aggiornamenti ogni mese, con condizini particolari per gli abbonati a ''TI-99 NEWSOFT'' (il secondo prezzo si riferisce a tali condizioni).
Tutti i prezzi sono comprensivi di IVA, ma escludono le spese di imballo e spedizione. Potrete effettuare il pagamento in contrassegno al ricevimento della merce. Per le ordinazioni, utilizzate la cartolina allegata alla rivista.

MD-14 MINI MEMORY PLUS
Mini Memory implementata con 4 KBytes di memoria RAM in più. 10 KBytes di GROM, 8 KBytes di RAM, batteria interna per il mantenimento dei dati a computer spento. Assemblatore su cassetta ''Assembler line by line'' fornito in dotazione. Possibilità di caricare direttamente in Mini Memory programmi in assembler molto elaborati con l'opzione ''RUN PROGRAM FILE'', come ''Defender'', che viene fornito in omaggio.
Manuale in italiano.

L. 160.000/145.000

RS-01: Interfaccia seriale più software su cassetta per la gestione delle comunicazioni via modem.
Uscita standard RS232 collegabile a tutti i modem in commercio.
Manuale in italiano per l'utilizzo dei vari protocolli di comunicazione.

L. 170.000/150.000

Offerte speciali solo abbonati

OS-01: TI LOGO Mondadori con manuale espansione di memoria 32K, libro ''Impariamo a programmare in Basic''.

L. 365.000

OS-02: Due joystick più cavo adattatore.

L. 58.000

OS-06: Espansione di memoria 32K più interfaccia parallela completa di cavo.

L. 350.000

OS-07: Extended Basic Grafico con manuale in italiano più espansione di memoria 32K.

L. 420.000

OS-08: Stampante EPSON LX-80 F/T più interfaccia parallela CENTRONICS più ''NEW WRITER'' (scegliere versione per disco o cassetta).

L. 1.050.000

NOVITÀ DEL MESE

TI-COMPIUMARKET

TI-WRITER

Elaboratore di testi professionale su cartuccia piu' disco. Dotato delle principali caratteristiche dei word processors per Personal Computer, consente il salvataggio e caricamento di testi da disco, l'inserimento, spostamento e copiatura di blocchi di testo, la stampa di un testo con numerazione automatica delle pagine, la stampa di mailing list con inserimento automatico degli indirizzi, la giustificazione del margine destro, la tabulazione, ecc.
Si possono inoltre inserire in ogni momento, codici di controllo della stampante, inclusi esponenti, pedici e tutti i caratteri speciali.
L'opzione "Text Formatter" permette inoltre di formattare nel modo voluto un testo precedentemente salvato su disco.
Il programma viene fornito con il manuale originale in inglese.

MD-13: TI-WRITER **L. 145.000 / 130.000**

NEW WRITER - Elaboratore di testi per TI 99/4A

Questo programma di word-processing, scritto in extended basic, consente la scrittura e l'editing del testo a tutto schermo e su un'unica riga continua, senza finestre laterali.
Il formato di stampa viene richiesto solo al momento della stampa del testo; il programma provvede automaticamente alla giustificazione del margine destro, spezzando le parole in sillabe secondo la grammatica italiana e inserendo il trattino di "a capo".
Durante la scrittura, e' possibile inserire dei codici di comando della stampante, per avere caratteri allargati, condensati, sottolineati, enfatizzati, ribattuti, ecc.
Il programma gira anche senza espansione di memoria, ma con l'aggiunta della 32K, il testo memorizzabile passa da una a quattro pagine stampate.
Il programma e' disponibile sia nella versione per disco che per registratore a cassette.

SC-15: NEW WRITER per disco L. 35.000/30.000
SC-16: NEW WRITER per cassetta L. 35.000/30.000

TI-99 COMPIÙMARKET

MD-12: EXTENTENDED BASIC GRAFICO. Il nuovo Extended Basic II plus della Mechatronic offre, oltre alle prestazioni della versione precedente, due nuove serie di istruzioni:
Più di 20 nuove istruzioni, che non sono ottenibili con la versione standard e che semplificano sensibilmente lo sviluppo del programma e la velocità di esecuzione. Ad esempio routine di hardcopy, accesso diretto alla VDP, spostamento di blocchi di memoria, save e load di blocchi di memoria, ecc.
Grafica ad alta risoluzione. Permette di indirizzare ogni singolo pixel dello schermo usando tutti i 16 colori, sia nel disegno che nello sfondo. 40 potenti istruzioni grafiche permettono di tracciare linee, cerchi, rettangoli, archi, diagrammi circolari, ecc., di salvare su disco la grafica e di ottenere hard copy. Le istruzioni grafiche sono utilizzabili solamente con l'espansione di memoria 32K.
L'Extended Basic II plus viene fornito completo di manuale in italiano dell'extended standard, più manuale dettagliato delle nuove istruzioni.
L. 295.000/265.000

MD-22: Implementazione Extended Basic. Se già possedete l'extended basic Mechatronic, potete dotarlo del nuovo set di istruzioni del modulo II plus spedendocelo insieme alla cartolina market; l'implementazione verrà effettuata inserendo all'interno del modulo le RAM fornite dalla Mechatronic. La modifica include il manuale con il nuovo set di istruzioni.
L. 100.000/90.000

Attenzione: questa modifica non è possibile sul modulo originale Texas Instruments.

SC-12: Programma su cassetta per grafica ad alta risoluzione. Se possedete l'extended basic originale Texas, potrete comunque disporre della grafica ad alta risoluzione del modello II plus, caricando questo programma, che viene fornito completo di manuale di istruzioni.
L. 65.000/55.000

Software

MD-03: Modulo Terminal Emulator.
L. 57.000/52.000

MD-05: Modulo ADD/SUB A/1 educativo.
L. 23.000/21.000

MD-06: Modulo MATH GAMES 6 educativo.
L. 23.000/21.000

MD-07: Modulo NUMERATION 1 educativo.
L. 23.000/21.000

MD-08: Modulo DEMOLITION DIVISION educativo.
L. 23.000/21.000

MD-09: Modulo MATH GAMES 2 educativo.
L. 23.000/21.000

MD-10: Modulo EARLY READING educativo.
L. 23.000/21.000

SC-01: Cassetta TI-99 CALC con manuale.
L. 39.000/35.000

SC-02: Cassetta Virgin ROBOPODS.
L. 19.000/17.000

SC-03: Cassetta Virgin FUNPAK.
L. 19.000/17.000

SC-04: Cassetta Virgin FUNPAK 2.

SC-05: Cassetta Virgin FUNPACK 3.
L. 19.000/17.000

SC-06: Cassetta DECATHLON
L. 15.000/13.000

SC-07: Cassetta BLEGRASS corsa ippica.
L. 13.000/12.000

SC-08: Cassetta GOBLIN REVENGE labirinto.
L. 13.000/12.000

SC-09: Cassetta SNOW TREK corsa di slitte.
L. 13.000/12.000

SC-10: Cassetta UP PERISCOPE battaglia navale.
L. 13.000/12.000

SC-11: Cassetta FAMILY GAME - 4 giochi.
L. 13.000/12.000

SC-13: Cassetta Oldies But Goodies (Games 1) 5 giochi.
L. 13.000/12.000

SC-14: Cassetta Personal Financial AIDS; pianificazione familiare.
L. 13.000/12.000

This Collection Is for all TI99/4A User. If like it please join at International TI99 forum present on site ATARIAGE.COM
Greetings from Italy. Astharot